HP 3000 Computer Systems

HEWLETT PACKARD

QUERY
reference manual

HP 3000 Computer System

QUERY

Reference Manual

5303 STEVENS CREEK BLVD., SANTA CLARA, CALIFORNIA, 95050

Part No. 30000-90042
Product No. 32216A

Printed in U.S.A. 6/76

NOTICE

The information contained in this document is subject to change without notice.

HEWLETT-PACKARD MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Hewlett-Packard shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance or use of this material.

Hewlett-Packard assumes no responsibility for the use or reliability of its software on equipment that is not furnished by Hewlett-Packard.

This document contains proprietary information which is protected by copyright. All rights are reserved. No part of this document may be photocopied, reproduced or translated to another program language without the prior written consent of Hewlett-Packard Company.

Copyright © 1976 by HEWLETT-PACKARD COMPANY

LIST OF EFFECTIVE PAGES

The List of Effective Pages gives the most recent date on which the technical material on any given page was altered. If a page is simply re-arranged due to a technical change on a previous page, it is not listed as a changed page. Within the manual, changes are marked with a vertical bar in the margin.

Pages	Effective Date
Title	June 1976
ii to ix	June 1976
1-1 to 1-12	June 1976
2-1 to 2-30	June 1976
3-1 to 3-25	June 1976
4-1 to 4-45	June 1976
5-1 to 5-20	June 1976
6-1 to 6-25	June 1976
A-1 to A-5	June 1976
B-1 to B-2	June 1976
C-1	June 1976
I-1 to I-2	June 1976

PRINTING HISTORY

New editions are complete revisions of the manual. Update packages, which are issued between editions, contain additional and replacement pages to be merged into the manual by the customer. The date on the title page and back cover of the manual changes only when a new edition is published. When an edition is reprinted, all the prior updates to the edition are incorporated. No information is incorporated into a reprinting unless it appears as a prior update. The edition does not change.

The software product part number printed alongside the date indicates the version and update level of the software product at the time the manual edition or update was issued. Many product updates and fixes do not require manual changes, and conversely, manual corrections may be done without accompanying product changes. Therefore, do not expect a one to one correspondence between product updates and manual updates.

First Edition	June 1976	32216A.00.00
Update No. 1	Apr 1978	32216A.03.05
Update No. 2	May 1979	32216A.04.01
Update No. 2 Incorporated	Apr 1979	32216A.04.01

PREFACE

This manual describes QUERY/3000, the Hewlett-Packard Data Base Inquiry Facility which is designed to run under the control of the HP 3000 Multiprogramming Executive (MPE) Operating System. QUERY/3000 enables you to access data in an IMAGE/3000 data base without writing a computer program.

In order to use QUERY you should know enough about the MPE Operating System to start a job or operate in session mode. It is recommended that you read the first section of *Using the HP 3000* if you are not an experienced MPE user. Knowledge of EDIT/3000 and simple MPE file commands is useful for some of the advanced QUERY techniques but is not required to operate QUERY.

You may need to consult the following manuals when using QUERY:

- *HP 3000 Computer System IMAGE Reference Manual* (32215-90003) gives complete information about the IMAGE/3000 Data Base Management System. (In the text this manual is abbreviated as *IMAGE/3000 Reference Manual.*)
- *Using the HP 3000* (03000-90121) provides a brief introduction to the MPE Operating System.
- *MPE Commands Reference Manual* (30000-90009) contains a complete description of the MPE/3000 Operating System commands.
- *SORT/3000 Reference Manual* (32214-90001) describes the SORT/3000 procedures.
- *DS/3000 Reference Manual* (32190-90001) contains a complete description of the Distributed Systems software.

This manual is arranged in six sections and three appendices. Section I introduces IMAGE/QUERY terminology, briefly describes an IMAGE data base, and discusses other important general IMAGE, QUERY, and MPE concepts. In Section II instructions are given for operating QUERY/3000 in session or job mode and the QUERY commands which specify the environment and assist you in using QUERY are described. The QUERY commands in the next three sections are organized by function. Section III describes commands which locate data and add, delete, or change data in the data base. Section IV discusses the various commands which report on data and Section V describes commands available for creating, altering, and displaying other QUERY commands saved in a file as procedures. All error messages are described in Section VI and Appendix C. Appendix A contains a command summary or quick reference table and Appendix B lists all ASCII characters. Methods of accessing a remote data base are described in Appendix D.

If your 3000 operating system is MPE-C, the following differences should be noted:

- The *IMAGE Reference Manual (30000-90041)* should be used.
- Whenever the *MPE Commands Reference Manual* is referenced in this manual, use the *MPE/3000 Operating System Reference Manual* (03000-90005).
- All references to R4 extended precision numbers should be translated to R3. The number of significant digits in an R3 number is 11 to 12 and the largest accurate absolute integer

is 549,755,813,887. QUERY rounds to 11 digits when printing R3 values. The minimum and maximum values for R3 numbers can be computed by rounding the R4 values in Table 1-1 to 11 significant digits.

- It is not possible to access a remote data base.

CONVENTIONS USED IN THIS MANUAL

NOTATION	DESCRIPTION
[ ]	An element inside brackets is *optional*. Several elements stacked inside a pair of brackets means the user may select any one or none of these elements. Example: [A / B] user may select A or B or neither
{ }	When several elements are stacked within braces the user **must** select one of these elements. Example: {A / B / C} user must select A or B or C.
italics	Lowercase italics denote a parameter which must be replaced by a user-supplied variable. Example: CALL *name* *name* one to 15 alphanumeric characters.
underlining	Dialogue: Where it is necessary to distinguish user input from computer output, the input is underlined. Example: NEW NAME? ALPHA1
superscript C	Control characters are indicated by a superscript C Example: Y^C
return	*return* in italics indicates a carriage return
linefeed	*linefeed* in italics indicates a linefeed
. . .	A horizontal ellipsis indicates that a previous bracketed element may be repeated, or that elements have been omitted.

CONTENTS

Section I — Page

INTRODUCING QUERY/3000

Passwords . . . 1-2
Structure of the Data Base . . . 1-2
- Fully-Qualified Data Item Names . . . 1-3
- Data Types . . . 1-4
- Data Values . . . 1-4
- Compound Data Items . . . 1-6
- Data Set Relations . . . 1-6

Sample Data Base . . . 1-6
Commands . . . 1-9
Modes of Access . . . 1-9
Logging and Mode Selection . . . 1-12
MPE Files . . . 1-12
QUERY/3000 Character Set . . . 1-13

Section II — Page

OPERATING QUERY/3000

Using QUERY in Session Mode . . . 2-1
Using QUERY in Job Mode . . . 2-4
ASSIGN . . . 2-5a
DEFINE . . . 2-6
DATA-BASE= . . . 2-8
PASSWORD= . . . 2-10
MODE= . . . 2-11
DATA-SETS= . . . 2-12
- Data Set Selection Rules . . . 2-12
- Automatic Data Set List Additions . . . 2-13

PROC-FILE= . . . 2-16
OUTPUT= . . . 2-18
EXIT . . . 2-22
FORM . . . 2-23
HELP . . . 2-28
SHOW . . . 2-29
VERSION . . . 2-30

Section III — Page

LOCATING AND UPDATING DATA

Using the FIND Command . . . 3-2
FIND . . . 3-3
- U and X Type Values . . . 3-4
- Logical Connectors . . . 3-4
- Multiple Values . . . 3-5
- Using Null Values . . . 3-6

FIND ALL . . . 3-9
FIND procedure . . . 3-10
FIND CHAIN . . . 3-11
UPDATE ADD . . . 3-14
- Using Quote Marks . . . 3-15
- Null Values . . . 3-15
- Real Number Values . . . 3-15a
- Detail Data Sets . . . 3-16
- Terminating UPDATE ADD . . . 3-16

UPDATE DELETE . . . 3-18
- Security Provisions . . . 3-18
- Master Data Set Entries . . . 3-18

UPDATE REPLACE . . . 3-20
- Session Mode . . . 3-21
- Job Mode . . . 3-21
- Null Values . . . 3-21

UPDATE procedure . . . 3-24

Section IV — Page

REPORTING

LIST . . . 4-2
- Listing Format . . . 4-4
- Listing a Subset of the Data . . . 4-5
- Determining the Data Set to be Listed . . . 4-6
- The Relation of LIST and FIND . . . 4-6

REPORT ALL . . . 4-10
REPORT . . . 4-13
- Designing a Report . . . 4-14
- Using Report Statements . . . 4-15
- Header Statements . . . 4-16
- Detail Statements . . . 4-18
- Edit Statements . . . 4-20
  - Alphanumeric Edit Mask . . . 4-20
  - Numeric Edit Masks . . . 4-21
  - Real Numbers . . . 4-21
- Sort Statements . . . 4-25
  - Control Breaks . . . 4-27
  - Major to Minor Sort Fields . . . 4-27
  - Maximum Number of Sort Items . . . 4-28
- Group Statements . . . 4-29
- Total Statements . . . 4-32
- Register Statements . . . 4-36
  - Using QUERY Registers . . . 4-37
  - Initializing Registers . . . 4-37
  - Register and Data Types . . . 4-38
  - Numeric Literals . . . 4-39
- Output Control Statements . . . 4-42

REPORT procedure . . . 4-44
- Skipping and Spacing . . . 4-45t

Section V — Page

USING PROCEDURES AND XEQ FILES

Using QUERY Commands from a File . . . 5-1
CREATE . . . 5-2
- File Input . . . 5-3
- Terminal Input . . . 5-3

CREATE SPACE . . . 5-6
DISPLAY . . . 5-7
- Displaying to a File . . . 5-7

DISPLAY LIST . . . 5-9
ALTER . . . 5-10
- Insert Statement . . . 5-11
- Replace Statement . . . 5-12
- Delete Statement . . . 5-13
- End Statement . . . 5-14

DESTROY . . . 5-15

CONTENTS

RENAME 5-16
XEQ 5-17

Section VI Page
QUERY/3000 MESSAGES 6-1

Appendix A
QUERY/3000 COMMAND SUMMARY A-1

Appendix B
ASCII CHARACTER SET B-1

Appendix C
EXCEPTIONAL ERROR CONDITIONS C-1

Appendix D
ACCESSING A REMOTE DATA BASE D-1

ILLUSTRATIONS

Title	Page
Data Base Structure	1-7
Sample Entry Values	1-7
IMAGE/QUERY Environments	1-11
Job Mode Operation	2-5
FORM ITEMS Output	2-24
FORM SETS Output	2-25
FORM PATHS Output	2-25
FORM data item name Output	2-26
FORM data set name Output	2-26
FORM Output	2-27
General Report Format	4-14
Sample Report	4-15
Sample Output Using Numeric Edit Masks	4-23
REPORT Procedure Named REP4	4-45
Billing Report From an XEQ File	5-20
Using Method 1	D-1
Using Method 2	D-2
Using Method 3	D-3

TABLES

Title	Page
Data Item Types	1-5
Command Categories	1-8
FIND Command Relational Operators	3-4
LIST Command Relational Operators	4-3
Field Widths	4-4
Negative Number Representation	4-11
REPORT Statements	4-13
Numeric Edit Mask Characters	4-22
Numeric Edit Mask Combinations	4-23
Statement Parameters	4-45
Commands Allowed as Procedures	5-1
General Messages	6-1
ALTER Command Messages	6-3
CREATE Command Messages	6-4
DATA-BASE= Command Messages	6-5
DATA-SETS= Command Messages	6-7
DESTROY Command Messages	6-7
DISPLAY Command Messages	6-7
FIND Command Messages	6-8
FORM Command Messages	6-11
LIST Command Messages	6-11
OUTPUT= Command Messages	6-13
PROC-FILE= Command Messages	6-14
RENAME Command Messages	6-15
REPORT Command Messages	6-15
UPDATE Command Messages	6-21
XEQ Command Messages	6-25

INTRODUCING QUERY/3000

SECTION I

QUERY/3000 provides a simple method of accessing an IMAGE/3000 data base without programming effort. You may use QUERY to do the following:

- enter data
- modify or delete data values on-line
- retrieve data which meets selection criteria
- report on the data retrieved.

You perform these operations by entering simple commands consisting of English-language key words such as FIND and REPORT.

You only need to know the relationships of the data base elements and not the structure of the disc files; QUERY finds the data and performs the operations in response to your commands using the data set and data item names you specify.

QUERY can be used from either a terminal or a batch input device such as a card reader. If you are in MPE session mode at a terminal, QUERY operates interactively, prompting you for commands, issuing error messages when an error occurs, and printing other information of interest to you. You can operate QUERY in batch mode as an MPE job from a batch input device. If a file system or IMAGE error occurs, QUERY prints an error message; in some cases, such as file read or write errors, QUERY terminates the job.

A sequence of commands can be stored in a file and executed at any time by entering a single command, XEQ, as part of a session or a job. A frequently used or complex command can be stored as an individual procedure in a command file (known as the Proc-file). The procedure name can then be used in place of the command parameters.

You may request information about the data base structure with the FORM command and information about QUERY commands, their function, format, and parameters with the HELP command.

QUERY adheres to all of the IMAGE/3000 security provisions described in the *IMAGE/3000 Reference Manual.* You must enter one of the valid passwords for the data base you are using. This word determines which information you may access in the data base.

QUERY's report formatting capability enables you to build reports with header and column labels, page numbers, and group labels. In addition, you can sort entries through multiple fields, as well as total, average, or count columns of numeric data values. QUERY also provides 10 registers for doing computations for reports.

If your local HP 3000 and a remote HP 3000 have Distributed Systems (DS/3000) capability, you can use QUERY to access a data base residing on the remote computer and retrieve information to be used on the local system. Three methods for accessing a remote data base are described in Appendix D.

All of the tasks described above can be accomplished without programming. If you are developing programs which access IMAGE data bases through the IMAGE library procedures, QUERY makes an excellent debugging aid. You can alter the data base content using your program and then use QUERY to examine the data and determine the results of your programmed changes.

PASSWORDS

Before you can access the data base, QUERY must know your password. Passwords are defined by the data base designer or administrator and control read and write access to information in the data base. You must ask the data base administrator for a password and enter the word as described later in this section.

It is possible to gain access to some data without a password if the data base is designed to allow such access.

STRUCTURE OF THE DATA BASE

It is not necessary to understand all the IMAGE features in order to use QUERY. However, it is important to have a general idea of the data base structure and to know the definitions of some IMAGE terms. You must use these terms in the QUERY commands described in the following sections. For those readers who are not familiar with IMAGE, here is an overview of the data base structure.

Each item of information in a data base is referenced by a *data item name.* The name associates the information with characteristics which describe it:

- the type of information (numeric or alphanumeric)
- its relation to other data in the data base
- the passwords required to read or write the information.

The data base designer organizes data items into *data sets* for the purpose of accessing them as a group. For example, an employee data set could contain the items EMPID, F-NAME, L-NAME, SOCSEC#, and SALARY. A credit union data set could contain EMPID, AMOUNT, TRANSCODE, and so forth. Each data set is referenced by a *data set name.*

Each time you enter a new employee's record into the data base, you can supply a value for each data item. For example,

F-NAME	SALLY
L-NAME	MERTON
SOCSEC#	527-58-6492
SALARY	18000.02

This group of values is stored as a single *data entry* in the data set.

QUERY allows you to locate particular entries which have values you specify. You can then change the values or print them in a report. You can also add or delete an entire entry if your password gives you the capability to write each data item in the data set.

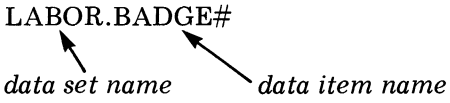
You can use a QUERY command, FORM, to display the names of each data set to which you have access and the names of the data items in these sets. Only the items to which you have at least read access are listed.

FULLY-QUALIFIED DATA ITEM NAMES

IMAGE allows the data base designer to use the same name for two or more items provided the items are not part of the same data set. If you are referring to such an item, you must specify which data set to access. You can do this in one of three ways:

- through the DATA-SETS=command described in Section II,
- through the DEFINE command DATA-SETS=prompt, which is also defined in Section II,
- by qualifying the data item name.

A fully-qualified data item name is a data set name followed by a period, followed by a data item name. For example,



BADGE# is the name of a data item in the data set named LABOR.

If BADGE# is also the name of a data item in a data set named EMPLOYEE, its fully-qualified name will be EMPLOYEE.BADGE#.

The discussion of data set lists in Section II explains more completely the way QUERY determines which data set to use.

DATA TYPES

The data base designer defines each data item as a particular type depending on what kind of information is to be stored in the item. It may be one of several types of integers, real or floating-point numbers, or ASCII character information. (Appendix B contains the ASCII character set.)

The FORM command also displays the data type for each item. When using QUERY, you will usually be unconcerned about the specific data type with these exceptions:

- when supplying values for an item, either to enter new information or to locate specific entries, you may want to know the acceptable range of values for a numeric type item
- when creating reports you should be aware of the item types in order to format the report properly
- when using the QUERY registers while printing a report, it is helpful to know how calculations affect the register values.

Detailed information about each of these situations is given with the appropriate command in the sections which follow. Table 1-1 contains a summary of the data types and the range of acceptable values for each type.

If you use QUERY to enter a value for a data item of type P (packed decimal), you should be aware of the way QUERY handles the sign of the value. A different code is used for the sign of a value entered with a plus sign than for a value entered without a sign. However, when QUERY retrieves unsigned and positive type P data items with the same value, they are considered to be equivalent. For example, +2 and 2 are equivalent. Data items with values +0 and 0 are also equivalent but a data item with a value of -0 is less than a data item with a value of +0 or 0. When using a data item value as a key, IMAGE differentiates between positive, unsigned, and negative data items with the same value.

DATA VALUES

When specifying the value of a particular data item, you must sometimes surround the value with quote marks. This type of value is called a *literal*. A character or string literal contains alphanumeric characters. For example,

"TANYA OAKLEY" "ZXR-93458273" "3215"

Character literals containing numeric values of the types listed under INTEGER or REAL in Table 1-1 are called, more specifically, numeric literals. For example,

"5468" "+408E-15" "-16.73892"

The rules for using quote marks are described with the specific commands which allow or require their use.

Table 1-1. Data Item Types

TYPE	MINIMUM	MAXIMUM
Integer		
I1	-32768	+32767
I2	-2,147,483,648	+2,147,483,647
J1	-9999	+9999
J2	-999999999	+999999999
K1	0	+65535
Z*n*	- (*n* digit number)	+ (*n* digit number)
P*n* (*n* must be even)	- (*n* - 1 digit number)	+ (*n* - digit number)
Note: *n* cannot exceed 255 and must be even for type P*n*. QUERY reports print at most 20 digits for type Z and 19 digits for type P data values.		
Real		
R2	-1.157921×10^{77}	$-0.863617 \times 10^{-77}$
	$+0.863617 \times 10^{-77}$	$+1.157921 \times 10^{77}$
	The largest accurate absolute integer is 8,388,607. (QUERY rounds to 6 digits when printing R2 values.)	
R4	$-1.157920892373162 \times 10^{77}$	$-0.8636168555094445 \times 10^{-77}$
	$+0.8636168555094445 \times 10^{-77}$	$+1.157920892373162 \times 10^{77}$
	The largest accurate absolute integer is 36,028,797,018,963,967 (QUERY rounds to 16 digits when printing R4 values.)	
Character		
U*n*	1 ASCII character (lower case not allowed)	*n* ASCII characters (lower case not allowed)
X*n*	1 ASCII character	*n* ASCII characters
Note: *n* cannot exceed 255. QUERY reports print at most 136 characters for type U and X data values.		

COMPOUND DATA ITEMS

IMAGE allows the data base designer to specify compound data items. These items occur more than once within the same data entry. Each occurrence of the data item is called a sub-item and each sub-item may have a value. QUERY locates and processes only the first sub-item with the FIND, UPDATE, LIST, and REPORT commands. REPORT ALL is an exception; it prints the values for all sub-items.

Even though QUERY updates only the first sub-item, it does preserve the existing values of all other sub-items.

DATA SET RELATIONS

There are three types of IMAGE data sets: *manual master*, *automatic master*, and *detail.* Master data sets are related to detail data sets through specific items called search (or key) items. The FORM command identifies the data set type and search items. You can carry out QUERY operations without reference to data set types or search items except when using the FIND CHAIN and UPDATE commands.

The data base designer can specify one or more sort items. These items are also identified by the FORM command. As a QUERY user, you need not understand the function of sort items, but you should be aware of which items are sort items when using the UPDATE command.

If you want to know more about data set relations and sort items, read the IMAGE/3000 Reference Manual description of the data base structure.

SAMPLE DATA BASE

Figure 1-1 and 1-2 illustrate a sample data base named STORE. The STORE data base is used in many examples in this manual. It contains six data sets (4 masters and 2 details):

- CUSTOMER contains information about each of the store's customers
- DATE-MASTER is an index of dates. It can be used to retrieve information by date from the SALES or INVENTORY data sets.
- PRODUCT contains information about each product in the store
- SALES has credit and purchase information
- SUP-MASTER contains information about each of the store's suppliers
- INVENTORY has product supply information

Both figures show a single entry for each data set. Figure 1-1 contains the data item names and figure 1-2 contains a sample of the values in one entry. The arrows in both figures illustrate the relationship of the data sets through search items. The four master data sets are shown in the center column and the details on the sides.

The data base is not meant to be a practical application but rather is designed to illustrate as many IMAGE/QUERY features as possible. It does illustrate some important design considerations. For a complete discussion of these see the *IMAGE/3000 Reference Manual.*

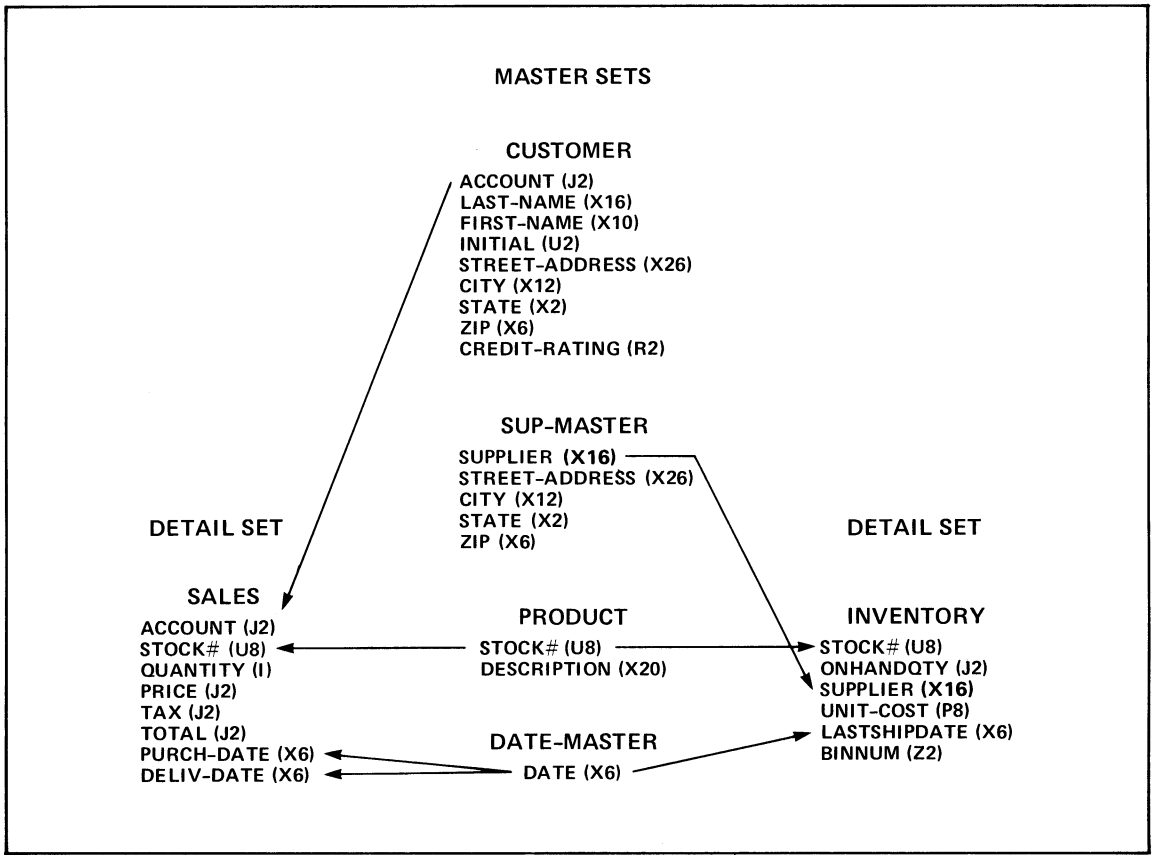


Figure 1-1. Data Base Structure

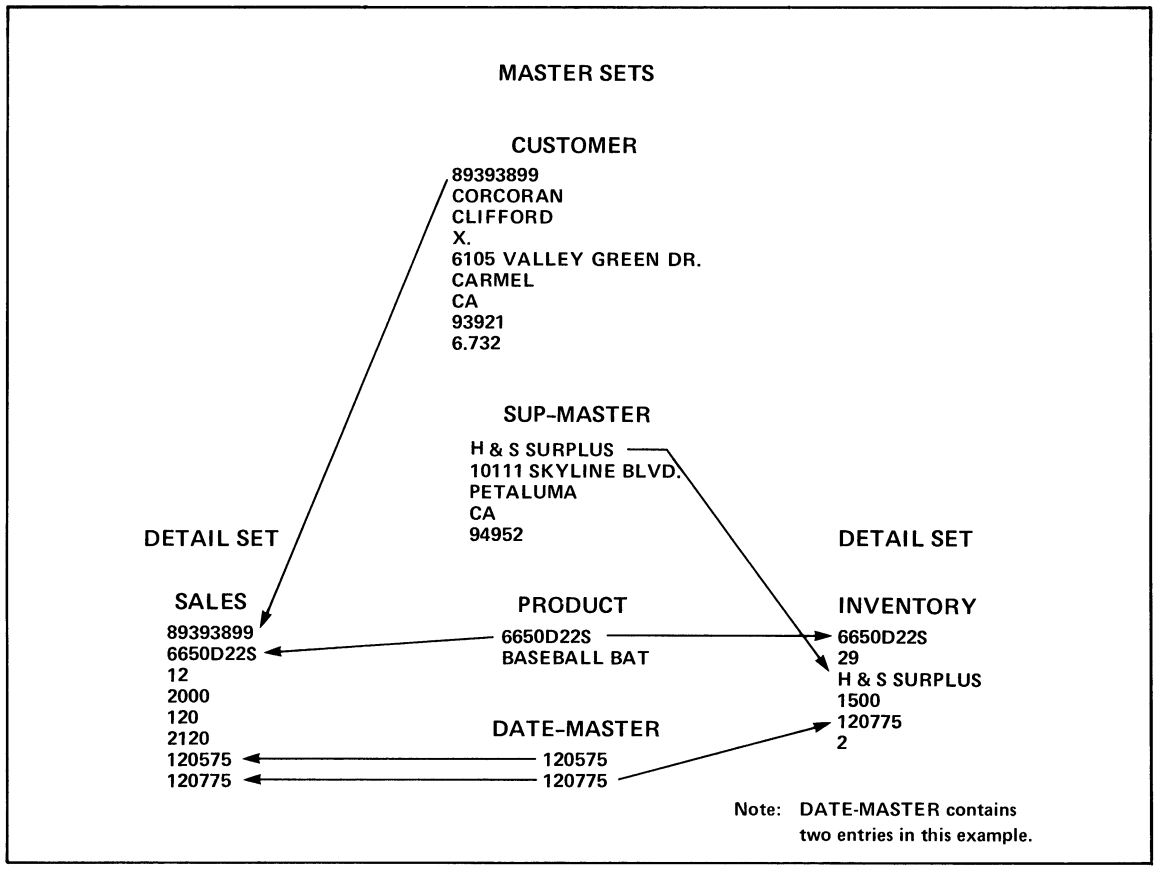


Figure 1-2. Sample Entry Values

Table 1-2. Command Categories

CATEGORY	COMMANDS	FUNCTION	SECTION
Environment	DATA-BASE=	Specifies the data base to be accessed	II
	DATA-SETS=	Specifies the data sets to be accessed.	
	PROC-FILE=	Specifies which Proc-file to be used.	
	OUTPUT=	Specifies the output device to be used for command output.	
	DEFINE	Indicates the current state of the Environment commands and prompts you for changes.	
	PASSWORD=	Specifies access password.	
	MODE=	Specifies access mode.	
Utility	FORM	Displays the structure of the data base being accessed.	II
	HELP	Lists a description of the QUERY command set.	
	EXIT	Terminates QUERY execution.	
	VERSION	Displays version, update, and fix information.	
Location	FIND	Locates data entries in the data base according to your specifications.	III
Updating	ADD or UPDATE ADD	Adds data entries to the data base.	
	DELETE or UPDATE DELETE	Removes data entries from the data base.	
	REPLACE or UPDATE REPLACE	Modifies data items in data entries.	
Reporting	REPORT	Reports about the entries located by the FIND command.	IV
	LIST	Prints entries with automatic formatting.	
Procedure	CREATE	Stores procedures in the Proc-file.	V
	DISPLAY	Displays procedures stored in the Proc-file.	
	ALTER	Modifies procedures currently stored in the Proc-file.	
	DESTROY	Removes procedures from the Proc-file.	
	RENAME	Changes procedure name.	
Operation	XEQ	Executes QUERY commands from a file.	

COMMANDS

QUERY commands are divided into seven categories according to their function. Table 1-2 describes the categories and the commands within each category. Detailed descriptions of each command are given in Sections II through V, however, you should understand the following characteristics which apply to all commands:

- Command names may be spelled out completely or abbreviated. The abbreviation for each command is specified with the format description. You must be careful to use the correct abbreviation.
- Commands consist of English keywords and parameters (both required and optional), all separated by one or more blanks.
- QUERY processes only the first 72 characters of a line (record). Any remaining characters can be used for comments or sequencing information.
- If the command you want to enter is longer than 72 characters, you may continue it on the next line by using an ampersand (&) as the last character on the current line. QUERY combines all lines connected with the & continuation character. Any blanks preceding the & are saved. Therefore, if you break a command name or other parameter with an &, the & should be adjacent to the last significant character with no intervening blanks.

Ampersands are not required to continue when entering the CREATE command in multiple lines (records).

MODES OF ACCESS

Each person using QUERY to access a data base must specify one of the available modes of access numbered from 1 to 8. Assuming that the MPE security provisions and your data base password permit it, you can do the following:

If your mode is	1	you may:	*find* (read), *replace*, *add*, *delete* entries (QUERY requests IMAGE dynamically lock and unlock the data base when accessing it).
	2		*find* and *replace* entries.
	3* or 4		*find*, *replace*, *add*, and *delete* entries.
	5		*find* entries. (QUERY locks and unlocks).
	6, 7*, or 8		*find* entries.

*Modes 3 and 7 give you exclusive access to the data base. All other modes allow others to share the data base. Search and sort items cannot be replaced.

A data base can only be shared in certain well-defined environments. The mode you specify must be acceptable for the environment already established by other IMAGE and QUERY users (if any) when you open the data base. Here is a summary of the acceptable environments:

- multiple mode 1 and mode 5 users
- multiple mode 6 and mode 2 users
- multiple mode 6 users and *one* mode 4 user

- multiple mode 6 and mode 8 users
- one mode 3 user
- one mode 7 user.

Subsets of these environments are also allowed. For example, there may be all mode 6 users or all mode 8 users. There may also be one mode 1 user or all mode 5 users and so forth.

If a mode 3 (or mode 7) user is currently accessing the data base that you want to access, you must wait until that user either terminates his or her IMAGE or QUERY session or accesses a different data base. This is true anytime you try to access a data base with a mode which is incompatible with other users accessing the same data base, unless you change your own mode of access.

Figure 1-3 illustrates the acceptable IMAGE/QUERY environments. Users who are smiling are allowed to operate IMAGE or QUERY simultaneously. The others must wait until only users with compatible modes are operating. Capability is abbreviated as follows: F = find, R = replace, A = add, and D = delete entries.

When deciding which mode to use, you should consider the following:

- If you merely want to find information and examine or report on it, you should open the data base with a find (read) only mode, thus allowing other users as much capability as you can tolerate. For example, if you open with:

  5 the data base is locked for some operations and may slow the rate of activity somewhat. Of course, if mode 1 or mode 5 users are already accessing the data base, you have no choice.

  6 mode 2 users can replace entries, one mode 4 user can replace, add, or delete entries, *or* mode 8 users can read entries while you are using the data base.

  8 no replacing, adding, or deleting of entries is allowed by other users.

  7 you are the only one accessing the data base.

An important advantage of using modes 5 through 8 is that the data base is opened for reading only. As a result, you are more likely to gain access to the data base by avoiding restrictions due to the MPE account structure. Also, the files are not marked for inclusion in the MPE system back-up tapes (SYSDUMP) since they are not altered in any way. This saves time when the daily system back-up procedure is executed.

- If you want to find information and replace data in existing entries but do not need to add or delete any entries (and do not want anyone else to add or delete entries), you should open the data base with mode 2.

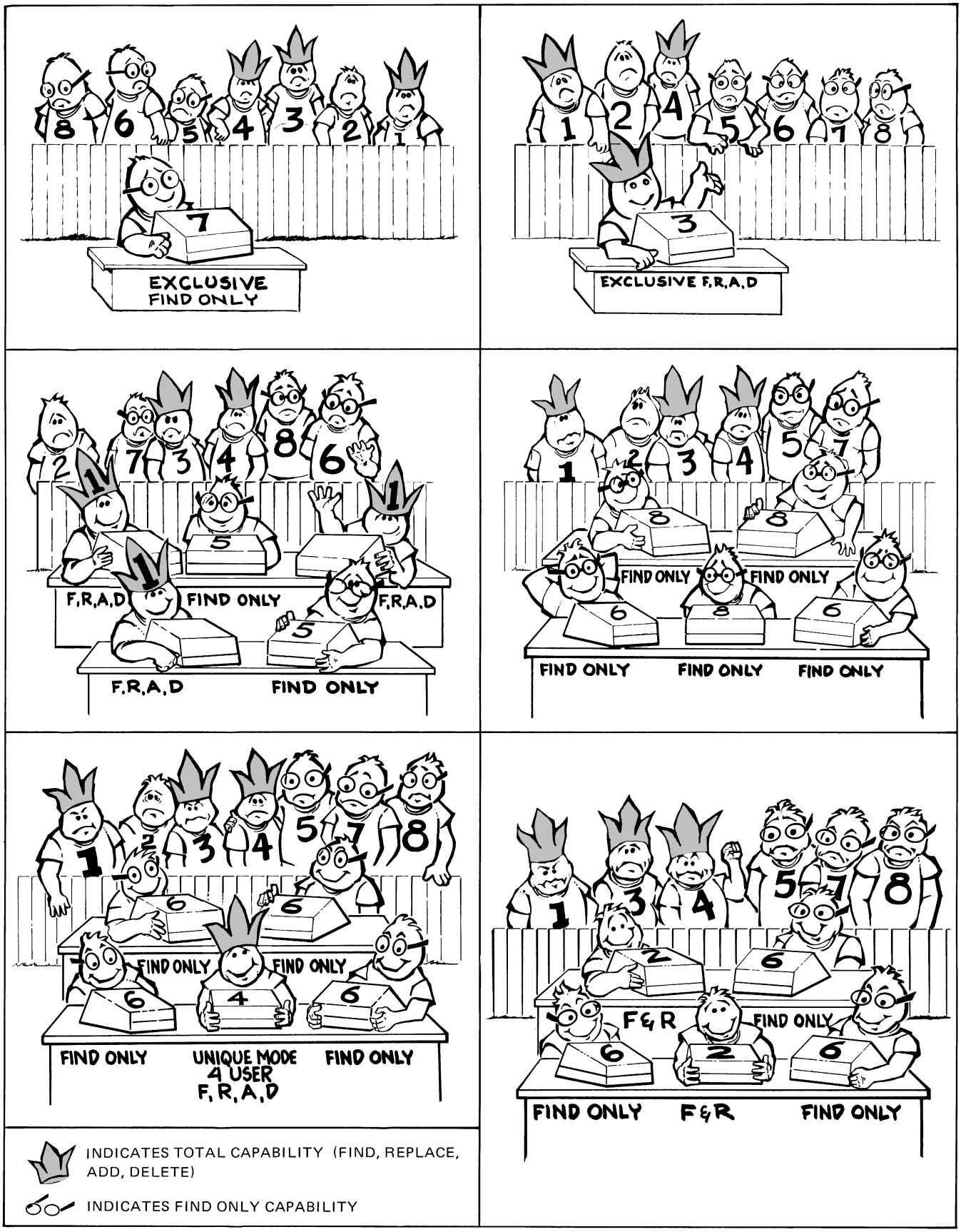


Figure 1-3. IMAGE/QUERY Environments

- If you want to perform all the operations including adding and deleting entries, you should open with one of the following modes:

  1 if you want QUERY to use lock and unlock while performing the operations and to allow other users to add and delete entries during your session. The comments for mode 5 above apply here also.

  4 if you want to have exclusive ability to change the data base but will allow mode 6 users to read while you are making the changes.

  3 if you want or must have exclusive access to the data base.

LOGGING AND MODE SELECTION

Opening the data base in one of modes 1 through 4 allows data base modifications to be logged to a logfile and subsequently recovered if there is a system failure. If the data base administrator has enabled the data base for logging, certain requirements of the logging system must be met before the data base can be accessed in one of modes 1 through 4.

These requirements are the responsibility of the data base administrator and console operator, and are discussed in the IMAGE Reference Manual. If any of these requirements are not satisfied, an IMAGE error message is returned.

MPE FILES

For those readers who are not familiar with the MPE Operating System, here is a brief introduction to some MPE concepts you may need to understand while using QUERY.

MPE treats input/output devices as files with standard names. For example, $STDLIST is the name of the standard output file for both session and job modes. A file is equated to a device class which may be a line printer, terminal, disc, or other peripheral device. $STDLIST is normally equated to a terminal in session mode and to a line printer in job mode. The system manager defines the various device classes and sets up the correspondence between device classes and standard files.

QUERY uses a file named QSLIST to allow you to alter the normal output device to any other output device. You must use an MPE :FILE command to equate QSLIST to the device class you select. (The default device class is "LP".) You then use the QUERY OUTPUT= command to change the output file from $STDLIST to QSLIST. The method for doing this is explained in detail with the OUTPUT= command in Section II. The system manager can tell you what the various device types on your system are.

Some QUERY commands accept input from MPE ASCII files. ASCII files contain ASCII characters (see Appendix B) and may be in card, disc, or magnetic tape format. You can use EDIT/3000 to create these files. See the terminal user's guide, *Using the HP 3000,* for instructions on how to create and edit files.

QUERY/3000 CHARACTER SET

The following ASCII characters are defined as the QUERY character set:

A through Z

a through z

0 through 9

+ –* / ? ' # & @ %

All other characters are referred to as "special" characters. When you encounter the term special characters in the following sections, it refers to all characters which are not part of this set. **Blanks are special characters.**

Lowercase characters are always upshifted (changed to uppercase) unless they are part of a literal or values for a data item of type X.

The complete ASCII character set is shown in Appendix B.

OPERATING QUERY/3000

SECTION II

QUERY can be run in either job (batch) or session (interactive) mode. In session mode, QUERY carries on a dialogue consisting of command prompts, error messages, and other messages of general interest to you. The manner in which QUERY proceeds depends upon your responses to QUERY messages and prompts. In job mode, QUERY reads commands and other input from the job input device. No dialogue occurs although QUERY does issue error and other messages informing you of action taken. It is your responsibility to anticipate the proper command and input sequence when operating in job mode.

When entering information in session mode, remember that you can delete one or more previously typed characters by entering a Control H character for each character you want to delete. You can delete the current line with Control X. Control characters are entered by pressing the Control key and holding it down while pressing the appropriate letter key.

USING QUERY IN SESSION MODE

To use QUERY in session mode, you must perform the following tasks:

```
return
:HELLO BROWN.DATAMGT
 SESSION NUMBER = #S56
WED, DEC 17, 1975, 10:49 AM
HP32000C.00.E1

:RUN QUERY.PUB.SYS

HP32216A.03.00 QUERY/3000   WED, DEC 17, 1975, 10:51 AM


QUERY/3000 READY
>
```

Press return. MPE prints a colon to indicate it is ready for a command. Log on to MPE using the HELLO command.

MPE prints information about your session and another colon.

Initiate QUERY execution with the RUN command.

QUERY responds with an opening message.

QUERY prints the "greater than" symbol when it is ready to accept a command.

If you do not understand the log-on procedure or MPE prints an error message, you should consult the first section of *Using the HP 3000* (see Preface for part number) which explains the procedure in detail.

Before you can actually access a data base you must specify a data base name, password, and access mode. Normally you will do this at the beginning of your QUERY session by using the DEFINE command or the DATA-BASE= command. You can use some QUERY

commands without identifying the data base as you will see when you become more familiar with all the QUERY commands.

The DEFINE command also allows you to define the procedure file, the output file (device), and the data set list. These parameters are all described in detail with the environment commands which follow in this section.

(2)

```
>DEFINE
DATA-BASE = >>STORE
PASSWORD = >>CLERK
MODE = >>1
DATA-SETS = >> return
PROC-FILE = >>MANPROC
OUTPUT = TERM
OUTPUT = >> return
```

or

```
>B=STORE
PASSWORD = >>BUYER
MODE = >>5
>P=PROCX
>
```

Define the QUERY operating environment as shown here or by using the individual environment commands as shown below. Both methods are explained later in this section. The DEFINE command causes prompts to be issued for each environment parameter.

QUERY prints two "greater than" symbols when prompting for command parameters or additional input.

Once the environment has been established, QUERY issues the > prompt again indicating that it is ready to accept a command.

(3)

```
>HALP
INVALID COMMAND
>HELP
```

Each command you enter is checked for correct form. If the command is incorrect, QUERY prints an error message and prompts you for another command. You may enter the correct version of the command or a different command.

```
QUERY COMMAND SET

ADD                ADDS DATA ENTRIES TO THE DATA BASE
ALTER              EDITS PROCEDURES IN THE PROC-FILE
CREATE             STORES PROCEDURES IN THE PROC-FILE
DATA-BASE          SPECIFIES DATA BASE
DATA-SETS          SPECIFIES DATA SETS
DEFINE             LISTS THE CURRENT STATE OF THE ENVIRONMENT
                   COMMANDS AND PROMPTS FOR CHANGES
DELETE             REMOVES DATA ENTRIES FROM THE DATA BASE
DESTROY            DELETES PROCEDURES FROM THE PROC-FILE
DISPLAY            DISPLAYS PROCEDURES IN THE PROC-FILE
EXIT               TERMINATES QUERY EXECUTION
FIND               LOCATES DATA ENTRIES IN THE DATA BASE
FORM               DISPLAYS THE STRUCTURE OF THE DATA BASE
HELP               LISTS A DESCRIPTION OF THE QUERY COMMAND SET
LIST               PRINTS DATA SET ITEM VALUES
  •                      •
  •                      •
  •                      •
```

(4)

```
>break key pressed
:FILE QSLIST;DEV=TAPE
:RESUME
READ PENDING
```

If you need to return control temporarily to the operating system, press the break key. Control transfers to MPE which prompts you for a command with a colon. To return control to QUERY, enter a RESUME command. MPE prints READ PENDING to tell you QUERY is waiting for a command. In this case QUERY does not print the > prompt.

(5)

```
< CONTROL Y >

>
```

If you want to terminate the execution of a QUERY command and return control to QUERY command mode, press Control Y. QUERY prints <Control Y> and prompts for a new command.

(6)

```
>EXIT

 END OF PROGRAM
:BYE

CPU (SEC) = 33
CONNECT (MIN) = 15
WED, DEC 17, 1975, 11:10 AM
END OF SESSION
```

Enter the EXIT command to terminate QUERY execution. Control returns to the operating system and you are prompted with a colon for an MPE command.

To terminate the session, enter the BYE command. The operating system reports the number of seconds of CPU time used as well as the number of minutes you were connected to the system.

USING QUERY IN JOB MODE

In job mode, QUERY reads commands and other input through the job input device (normally a card reader). All output goes to the device designated as $STDLIST, which is normally a line printer. Only the first 72 characters of each record are processed; the remaining characters in the record are ignored. QUERY commands consisting of more than 72 characters can be continued on following records by entering an & as the last non-blank character in the record to be continued.

It is your responsibility to anticipate the proper order of input. If a command is out of sequence or incorrectly entered, QUERY issues an error message. A file read or write error causes QUERY to terminate the job.

It is also your responsibility to specify the data base to be accessed using the DATA-BASE= command, and the procedure file to be used through the PROC-FILE= command. Figure 2-1 shows a card deck used to operate QUERY in job mode. The QUERY environment commands are described in detail later in this section.

If QUERY is unable to open the data base for any reason, it automatically terminates.

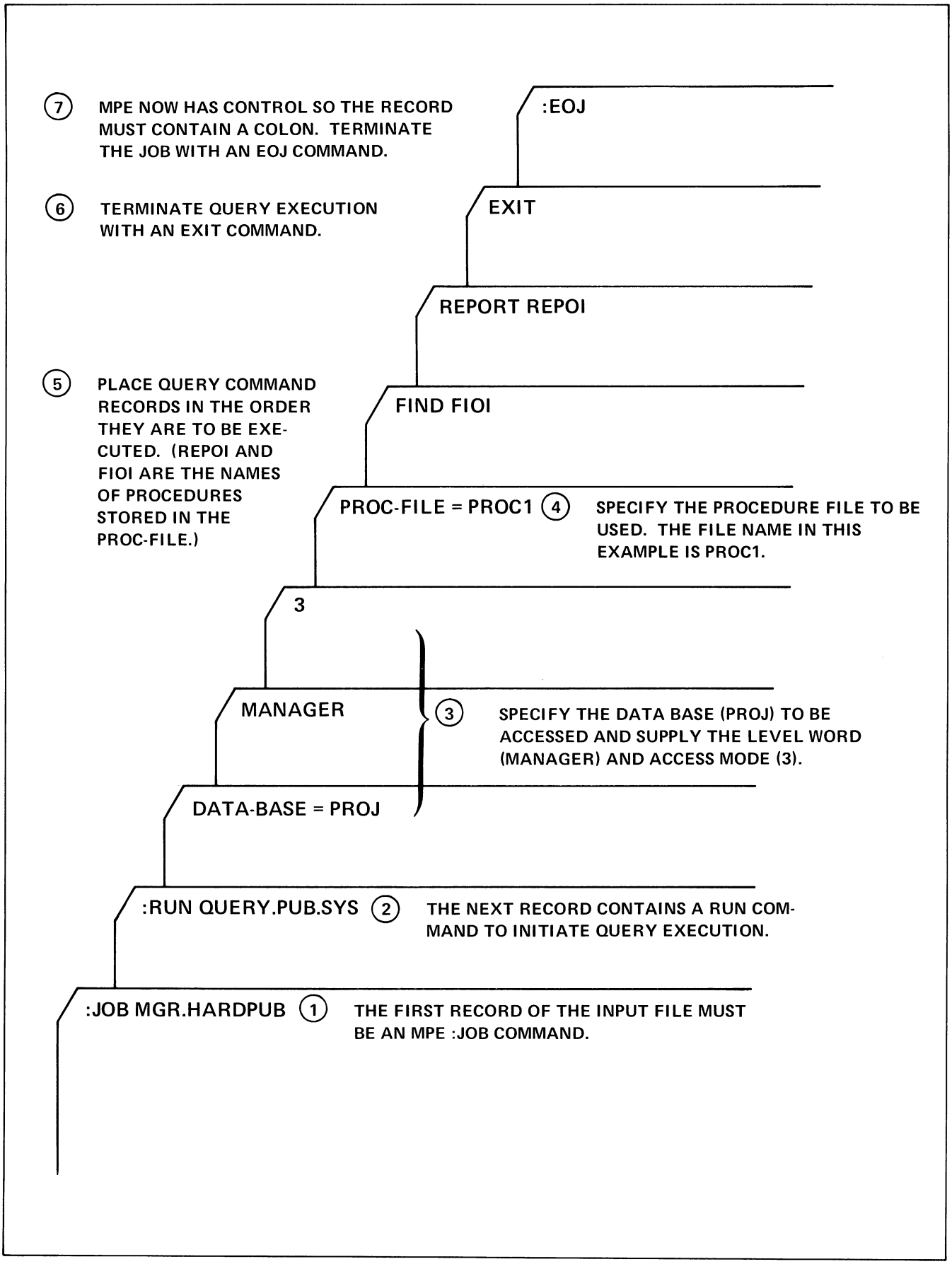


Figure 2-1. Job Mode Operation

Allows a user to disable or enable an option. The only option currently supported by the ASSIGN command is LOCKOPTION.

The form of the ASSIGN command is

{ ASSIGN / AS } *option* = { ON / OFF }

For example:

```
ASSIGN LOCKOPTION = OFF
       option
```

where *option* is the name of an option whose default state is "on" but can be set to "off". It may subsequently be reset to "on" by the user (currently applies only to LOCKOPTION).

When LOCKOPTION is on (default state), QUERY always locks any data base opened in mode 1 or 5 while any FIND and REPORT commands are executing in order to ensure that the resulting report does not contain errors due to dynamic changes made by other users concurrently accessing the data base. In some environments, this may be overprotective if you are sure that no other user is concurrently modifying the data pertinent to your FIND and REPORT commands. For example, protection may be provided through exclusive assignment of passwords or through exclusive scheduling of tasks. In such cases, automatic locking by QUERY can needlessly increase response time for other users. By setting the LOCKOPTION to "off", you inform QUERY not to lock any data base opened in mode 1 or 5.

The LOCKOPTION can be set off or on at any time and remains in effect until it is reset by a subsequent ASSIGN command. To determine the current state of the LOCKOPTION, use the SHOW command.

DEFINE

Lists the status of all of the Environment commands and enables you to change the environment specifications.

The form of the DEFINE command is

```
{DEFINE}
{DEF   }
```

When you enter the DEFINE command in either session or job mode, QUERY lists the state of each of the Environment commands and prompts you for changes. After the prompt, you may enter a new parameter for the command, or press *return* to maintain the current value.

When DEFINE is entered in job mode, you must anticipate the order of QUERY prompts and enter the new values in the proper order on the records immediately following the DEFINE command. A blank record indicates no change in value (like *return* in session mode).

The prompts (shown without replies) are:

```
DATA-BASE = >>
PASSWORD = >>
MODE = >>
DATA-SETS = >>
PROC-FILE = >>
OUTPUT = >> TERM
OUTPUT = >>
```

When you first use the DEFINE command after initiating QUERY execution, none of the command settings will be listed except OUTPUT=TERM. If you do not want to define a particular environment parameter respond with *return*. If you are only doing procedure maintenance you need not specify a data base. On the other hand, if you are not using procedures you do not have to define a Proc-file.

See the command descriptions which follow for details about the purpose of each environment specification.

EXAMPLE

The DEFINE command can be used to set up the environment for your QUERY session.

```
QUERY/3000 READY

>DEFINE
DATA-BASE = >>STORE
PASSWORD = >>CLERK
MODE = >>5
DATA-SETS = >> return
PROC-FILE = >>MANPROC
OUTPUT = TERM
OUTPUT = >> return
```

QUERY prompts for the data base name, the password, mode, data set list, and procedure file name. The current output device is the terminal. If you do not want to change it, reply with return.

The command is also useful to list the current environment and change it.

```
>DEFINE
DATA-BASE = STORE
DATA-BASE = >> return
PASSWORD = CLERK
PASSWORD = >> return
MODE = 5
MODE = >>return
DATA-SETS = >> return
PROC-FILE = MANPROC.IMAGE.DATAMGT
PROC-FILE = >>CANNED
OUTPUT = TERM
OUTPUT = >>return
>
```

Once the environment has been defined, QUERY prints the current setting for each environment command and allows you to change it or press return to leave it as is.

The Proc-file is changed from MANPROC to CANNED.

DATA-BASE=

Specifies the data base which is to be accessed.

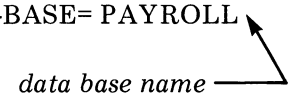
The form of the DATA-BASE= command is

{ DATA-BASE= } *data base name*
{ B= }

For example,

```
>DATA-BASE= PAYROLL
```

data base name



where *data base name* is the name of an IMAGE/3000 data base.

The data base may reside in any group or account on the local HP 3000 or a remote HP 3000, as long as you are allowed access to it through the MPE file security. To specify a data base that does not reside in your group and account, you must use a fully-qualified name in the form: *data base name.group.account.* For example, STORE.PUB.SYS is a data base named STORE in the PUB group of the SYS account.

If you want to access a remote data base, you may specify the name of a data-base-access file as the data base name. A data-base access file contains the name of a remote data base and the commands to establish a communications link and remote session with another HP 3000 computer.

Refer to the *IMAGE Reference Manual* for more information about how to create data-base-access files and to Appendix D for a description of available methods for accessing a remote data base using QUERY.

This command is used to specify which data base is to be accessed. You must use either this command or the DEFINE command to identify the data base you want to use before you can use any QUERY commands which access a data base. You may change the data base you are accessing at any time.

When you enter this command, QUERY prompts you for a password with the message:

PASSWORD=>>

Passwords are created by the data base administrator or designer who will tell you which one you may use. The password determines which data items you are allowed to read or write in the data base.

After you enter an appropriate response, QUERY prompts you for the desired access mode with the message:

MODE=>>

You must enter a valid access mode represented by a number between 1 and 8. The description of modes in Section I will help you determine which mode to use.

QUERY checks both prompt responses for validity and issues an error message if any of the answers is incorrect.

PASSWORD= and MODE= may also be used as commands. Descriptions of these commands follow.

If the DATA-BASE= command is entered in batch mode, the password and the mode number must follow the command in the next two records, respectively. If no password is required for the data base, the record following the command must be blank.

When you enter this command, QUERY first closes the current data base before attempting to open the new data base requested in the command. Therefore, if QUERY is unable to open the requested data base for some reason, you must specify a different data base to be accessed through the DATA-BASE= or the DEFINE command or terminate QUERY.

EXAMPLES

```
>DATA-BASE=STORE
PASSWORD = >>BUYER
MODE = >>3
>
```

Specify access to the STORE data base. Provide a password, BUYER, and a mode, 3, requesting exclusive read and write access to the data base.

```
>B=STORE.IMAGE
PASSWORD = >>CLERK
MODE = >>5
DATA BASE OPEN IN ANOTHER MODE

>B=STORE.IMAGE
PASSWORD = >>CLERK
MODE = >>6
>
```

Specify the data base STORE.IMAGE and password CLERK. Mode 5 is not allowed because other users are in incompatible modes. Mode 6 is used instead.

PASSWORD=

Changes your password.

The form of the PASSWORD= command is

{ PASSWORD= } *password*
{ PA= }

For example,

```
>PASSWORD= BOSS
```

password

where *password* is the word you have been given by the data base administrator to use when accessing the data base.

You may change your password, and thus, the group of data to which you have access. QUERY first closes the current data base before attempting to open the data base with the new password.

EXAMPLE

```
>PAS=CLERK
>AD INVENTORY
ILLEGAL ACCESS
>PAS=DO-ALL
>AD INVENTORY
STOCK#              =>>
```

Specify CLERK password.
A request to add an entry to the INVENTORY data base is rejected since CLERK does not allow write access to it. If the password is changed to DO-ALL, QUERY allows the user to add to the INVENTORY data set.

Changes the mode of access to the current data base.

The form of the MODE= command is

{ MODE= } *mode number*
{ M= }

For example,

MODE= 3

mode number

where *mode number* is a number (an integer) in the range of 1 to 8 representing the access mode you want to use.

You may use this command to change your mode of access to the data base. If the MODE= command is entered, QUERY first closes the current data base before attempting to open the data base with the new access mode.

EXAMPLE

```
>MODE=6
>AD INVENTORY
ILLEGAL ACCESS
>MODE=3
>AD INVENTORY
STOCK#          =>>6650D22S
ONHANDQTY       =>>11
SUPPLIER        =>>H & S SURPLUS
UNIT-COST       =>>1395
LASTSHIPDATE    =>>121575
BINNUM          =>>3

STOCK#          =>>//
>
```

A mode of 6 is specified. Since mode 6 does not allow write access to the data base, the user must change to a mode which allows such access and then QUERY allows the addition of an INVENTORY entry.

DATA-SETS=

Informs QUERY which data set to access in the event that a data item name which appears in more than one data set is used in a FIND or LIST command.

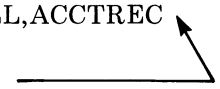
The form of the DATA-SETS= command is

{ DATA-SETS= } *[data set list]*
{ S= }

For example,

DATA-SETS= PAYROLL,ACCTREC

data set list



or

DATA-SETS=

where *data set list* is a list of data sets you want to access, each data set name separated from the next one by a comma. If *data set list* is not included, the set list is cleared.

In an IMAGE/3000 data base, the same name may be used for different data items if each appears in a different data set. As mentioned in Section I, you may use a "fully-qualified" data item name (*data set name.data item name*) to tell QUERY exactly which data item you want to access. However, if you are referring to several item names which appear in multiple data sets, a command could become quite lengthy. The DATA-SETS= command may save you some effort.

The DATA-SETS= command defines a list of one or more data sets. When you use a FIND or LIST command and QUERY encounters a data item name which appears in more than one data set but is not "fully-qualified", it checks the data set list to help resolve the ambiguity.

DATA SET SELECTION RULES

In session mode, if you reference a data item appearing in more than one data set, QUERY resolves which data set to use according to the following rule:

- If one (and only one) of the data sets containing the data item appears in the data set list, QUERY automatically uses that data set. Otherwise QUERY prompts you to supply the desired data set with the message:

data item name IS A MEMBER OF THESE SETS:

data set name, data set name, ...

WHICH SET DO YOU WISH TO USE?

>>*data set name*

You must type the name of the data set you want to access. If the name you provide does not match the names listed, QUERY repeats the prompt. If you decide you do not want to access the listed data sets, you may abort the command by pressing *return* instead of entering a data set name. You will be prompted for another command.

In job mode, QUERY cannot prompt you for the desired data set in the event there is ambiguity. The data set to be accessed is chosen according to the following rules:

- If exactly one data set containing the data item appears in the data set list, QUERY uses that data set.
- If more than one of the data sets containing the data item appears in the data set list, QUERY uses the last data set mentioned in the list.
- If no data set containing the data item is in the data set list, QUERY accesses the last data set (containing the data item) defined in the data base schema to which you have access. (The schema is the source description of the data base prepared by the data base designer.) You may want to use the FORM *data item name* command in session mode before you prepare your job to determine the order in which data sets containing the item appear in the schema. You can then decide what to include in the data set list.

In job mode, QUERY always informs you which data set was chosen (in case of ambiguity) with this message:

data item name IS A MEMBER OF THESE SETS:

data set name, data set name, ...

data set name USED

AUTOMATIC DATA SET LIST ADDITIONS

If a FIND or LIST command is executed which contains an unqualified data item name (i.e., a data item without a preceding data set name), QUERY automatically adds the name of the accessed data set to the data set list. This occurs whether the named data item appears in more than one data set or not.

To avoid any ambiguity when the same data item appears in multiple data sets, you can either:

- use fully-qualified data item names in all commands, or
- always reset the data set list (using the DATA-SETS= command) prior to entering a command using a data item which appears in multiple sets.

DATA-SETS=

EXAMPLES

```
>SETS=MUD,INVENTORY,SALES
ILLEGAL DATA SET NAME MUD
>DEF
DATA-BASE = STORE.IMAGE
DATA-BASE = >> return
PASSWORD = DO-ALL
PASSWORD = >> return
MODE = 1
MODE = >> return
DATA-SETS = INVENTORY,SALES
DATA-SETS = >> return
PROC-FILE = >> return
OUTPUT = TERM
OUTPUT = >> return
```

Enter the data set names you want in the list. If a name is invalid an error is printed.

If you use the DEFINE command, you can see which sets are currently in the list. The valid data set names are now in the list.

The next example illustrates how QUERY uses the data set list, clears it, and enters data set names automatically.

```
>DEFINE
DATA-BASE = STORE.IMAGE
DATA-BASE = >>return
PASSWORD = DO-ALL
PASSWORD = >>return
MODE = 1
MODE = >>return
DATA-SETS = SALES
DATA-SETS = >>return
PROC-FILE = MANPROC.IMAGE.DATAMGT
PROC-FILE = >>return
OUTPUT = TERM
OUTPUT = >>return
>FIND STOCK#=6650D22S
2  ENTRIES QUALIFIED
>DATA-SETS=
>FIND STOCK#=6650D22S
STOCK#              IS A MEMBER OF THESE SETS:
PRODUCT, SALES, INVENTORY
WHICH SET DO YOU WISH TO USE?
>>INVENTORY
5  ENTRIES QUALIFIED
>DEFINE
DATA-BASE = STORE.IMAGE
DATA-BASE = >>return
PASSWORD = DO-ALL
PASSWORD = >>return
MODE = 1
MODE = >>return
DATA-SETS = INVENTORY
DATA-SETS = >>return
PROC-FILE = MANPROC.IMAGE.DATAMGT
PROC-FILE = >>return
OUTPUT = TERM
OUTPUT = >>return
>FIND STOCK#=7391Z22F
2  ENTRIES QUALIFIED
>FIND SALES.STOCK#=7391Z22F
8  ENTRIES QUALIFIED
```

The data set list currently contains the SALES data set name.

When searching for entries with STOCK#=6650D22S, the SALES data set is automatically used since it is in the set list.

If the set list is cleared, QUERY asks which set you want to use.

INVENTORY was automatically placed in the data set list when specified in the FIND command above.

Now QUERY automatically searches the INVENTORY data set unless the data item name is qualified with another data set name as shown here with SALES.

PROC-FILE=

Specifies the name of the current procedure file.

The form of the PROC-FILE= command is

{ PROC-FILE= / P= } *filename* *[,n]*

For example,

```
PROC-FILE=FILEP, 50
```

filename *n*

where *filename* is the name of an MPE ASCII file. The file may reside in any group or account, as long as you are allowed access to it through the MPE file security. To specify a file that is not local to your group and account, you must use a fully-qualified file name in the form: *file.group.account*. For example, SPEC.PUB.SYS is a file named SPEC in the PUB group account of the SYS account.

n is the number of records in the file. The file may be from 5 to 400 records in length depending upon the number and length of procedures to be stored. (The default value is 126.) Each procedure is stored on a record boundary, and occupies one or more records.

The file used to store FIND, REPORT, and UPDATE commands as procedures is known as the Proc-file. Before using any of the QUERY procedure commands, you must specify the name of the Proc-file. This definition stays in force until changed (using PROC-FILE= or DEFINE) or until execution terminates.

If the file name does not exist, QUERY issues a message and opens and saves a disc file of size *n* with the specified file name using a file code of 1070.

If *n* is specified for an existing file, it is ignored.

Once a Proc-file has been declared, QUERY always uses that file when executing any of the following commands:

CREATE
ALTER
DISPLAY
DESTROY
RENAME
FIND *procedure name*
UPDATE *procedure name*
REPORT *procedure name*

If you have not specified a Proc-file before entering one of these commands, QUERY prints an error message.

EXAMPLES

```
>PROC-FILE=MANPROC
>DISPLAY LIST

F1          LP3          UPD1          REP4          ZAP

>P=CANNED
>DISPLAY LIST

XXX          YYY          ZZZ
```

MANPROC contains the listed procedures.

CANNED contains three procedures.

```
>PROC-FILE=PROCX
FILE DOES NOT EXIST, BEING CREATED
>
```

You can create a Proc-file with the PROC-FILE= command. If you do not specify the number of records QUERY creates a file with 126.

OUTPUT=

Selects the output device for the FORM, HELP, DISPLAY, LIST, REPORT, and VERSION commands.

The form of the OUTPUT= command is

```
{ OUTPUT= }     { TERM }
{ O=      }     { LP   }
```

For example,

OUTPUT=LP

or

OUT= TERM

where	TERM	indicates that output is to be displayed on the device specified in the MPE operating system as $STDLIST. ($STDLIST is normally a terminal in session mode and a line printer in job mode.)
	LP	indicates that output is to be sent to the file designated as QSLIST. QSLIST is equated to the MPE device class LP (line printer) unless you use an MPE :FILE command to specify otherwise. See below.

The default output device for the commands listed above is $STDLIST unless you use this command to change it. You can also use this command to change the output device back to $STDLIST at any time after setting it to QSLIST.

If you want to associate the file named QSLIST with a device other than the line printer you must use the MPE FILE command and then set OUTPUT=LP.

The form of this command is:

:FILE QSLIST; DEV= *device type*

where *device type* is a name referring to a class of devices or a specific device. The different device classes and names are set up by the system manager at the time the MPE operating system is configured. You must ask the system manager which name to use for the device you want to associate with QSLIST.

QUERY will not output a record longer than 136 bytes (characters) even though the maximum record length for the device may exceed this limit.

An example of a :FILE command used to divert output to a magnetic tape is:

```
:FILE QSLIST;DEV=TAPE
```

In this case, TAPE is the name of the class of devices known as magnetic tapes.

If you want to change the device associated with QSLIST while operating QUERY in session mode, you must use the *break* key or terminate QUERY and return to the MPE operating system before entering the :FILE command or enter it before running QUERY.

If you are operating in job mode, you may place a :FILE command in your job preceding the :RUN QUERY command.

All error messages and QUERY prompts are always sent through the $STDLIST device, regardless of whether OUTPUT=TERM or OUTPUT=LP.

EXAMPLES

QUERY is running in session mode. QSLIST is equated to a terminal initially.

```
>OUTPUT=LP                          Change output device to QSLIST.
>LIST INVENTORY                     The INVENTORY data set is listed on the
>OUT=TERM                           line printer.
>LIST INVENTORY                     Set output device to terminal again.
                                    The INVENTORY data set is listed on the
                                    terminal.

STOCK#      ONHANDQTY  SUPPLIER          UNIT-COS  LASTSH  BIN

6650D22S         5306  ACME WIDGET          1427  120375    3
2457A11C     11001345  ACME WIDGET          5031  120175    1
3586T14Y          144  CARDINAL MILLS        249  112075    2
```

OUTPUT=

The next example illustrates the method for sending output to a tape device. In this case, a procedure from the Proc-file is listed first on the terminal and then on a tape.

```
>DISPLAY FNAMES
```

List a procedure named FNAMES.

```
PROCEDURE: FNAMES

 001   FIND LAST-NAME="" END
```

```
> break key is pressed
:FILE QSLIST;DEV=TAPE
:RESUME
READ PENDING
OUTPUT=LP
>DISPLAY FNAMES
>OUTPUT=TERM
>
```

Press the break key and equate QSLIST to a magnetic tape device. Resume QUERY execution. Change the output device to QSLIST. Write the procedure to the QSLIST device. Set the output device to the terminal again.

In this example, QUERY locates a specific entry. The same procedure is followed as above except QSLIST is equated to a disc file named SAVER which MPE creates.

```
>FIND CUSTOMER.LAST-NAME IS CORCORAN END
USING SERIAL READ
1  ENTRIES QUALIFIED
>REPORT ALL
```

Retrieve data with FIND command.

```
ACCOUNT            =54283540
LAST-NAME          =CORCORAN
FIRST-NAME         =CLIFFORD
INITIAL            =C
STREET-ADDRESS     =6105 VALLEY GREEN DR.
CITY               =CARMEL
STATE              =CA
ZIP                =93921
CREDIT-RATING      = 7.10000
```

Report on data.

```
> break key is pressed
:FILE QSLIST=SAVER,NEW;DEV=DISC
:RESUME
READ PENDING
OUTPUT=LP
>REPORT ALL
>EXIT
```

Press the break key and define a new file named SAVER. Then resume QUERY execution. Use OUTPUT=LP and REPORT ALL commands to write data in the SAVER file.

EXIT

Terminates QUERY execution.

The form of the EXIT command is

```
{EXIT}
{E   }
```

You may enter the EXIT command whenever QUERY prompts for a command. QUERY execution terminates and control is returned to the operating system. The operating system then prompts you for a command with the colon (:) prompt character.

You must use an EXIT record to terminate a set of QUERY commands entered as a job. The EXIT record should be followed by an MPE command record (usually :EOJ).

EXAMPLE

```
>EXIT

 END OF PROGRAM
:BYE

CPU (SEC) = 33
CONNECT (MIN) = 15
TUE, JAN 13, 1976,  9:02 AM
END OF SESSION
```

FORM

Lists information about the data base currently being accessed.

The form of the FORM command is

{ FORM / FO } [*data set name* / *data item name* / SETS / ITEMS / PATHS]

For example,

FORM

or

FO PATHS

or

FORM DSET1

data set name

or

FO FNAME

data item name

where

data set name	is the name of a data set in the data base currently being accessed.
data item name	is the name of a data item in the data base currently being accessed.
SETS	tells QUERY to list information about each data set in the data base to which you have access.
ITEMS	tells QUERY to list information about each data item in the data base to which you have access.
PATHS	tells QUERY to list the relationship between data sets in the data base to which you have access.

FORM

FORM provides information about the current data base. The information contains only the names of data sets and data items to which you have at least read access. No other data sets and data items are listed. If a data base is not currently defined, QUERY issues a message and prompts you for another command.

If OUTPUT=TERM, the listing is sent to the standard list device for the job or session. If OUTPUT=LP, the listing is sent to the file named QSLIST. See the OUTPUT= command for information about QSLIST.

Figures 2-2 through 2-7 illustrate the output resulting from each FORM command option. If you enter FORM *name* and *name* refers to both a data item and a data set, the data set information is listed. If a data set or a data item has the name SETS, ITEMS, or PATHS, it is treated as a keyword parameter when used in a FORM command.

The FORM PATHS output shown in figure 2-4 lists the detail data sets associated with each master data set and the master data sets associated with each detail. It also lists the detail set item which is used as a key (search item name) and the detail set item which is used for sorting (if any).

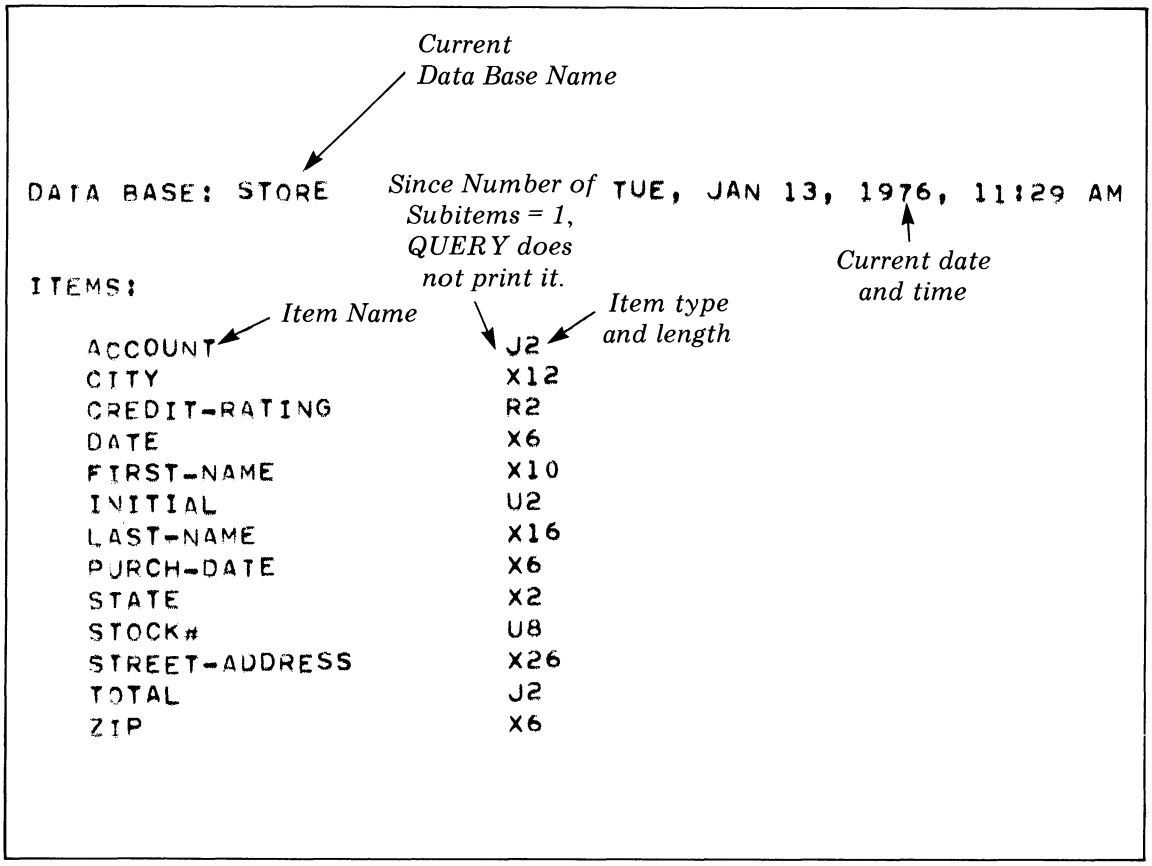


Figure 2-2. FORM ITEMS Output

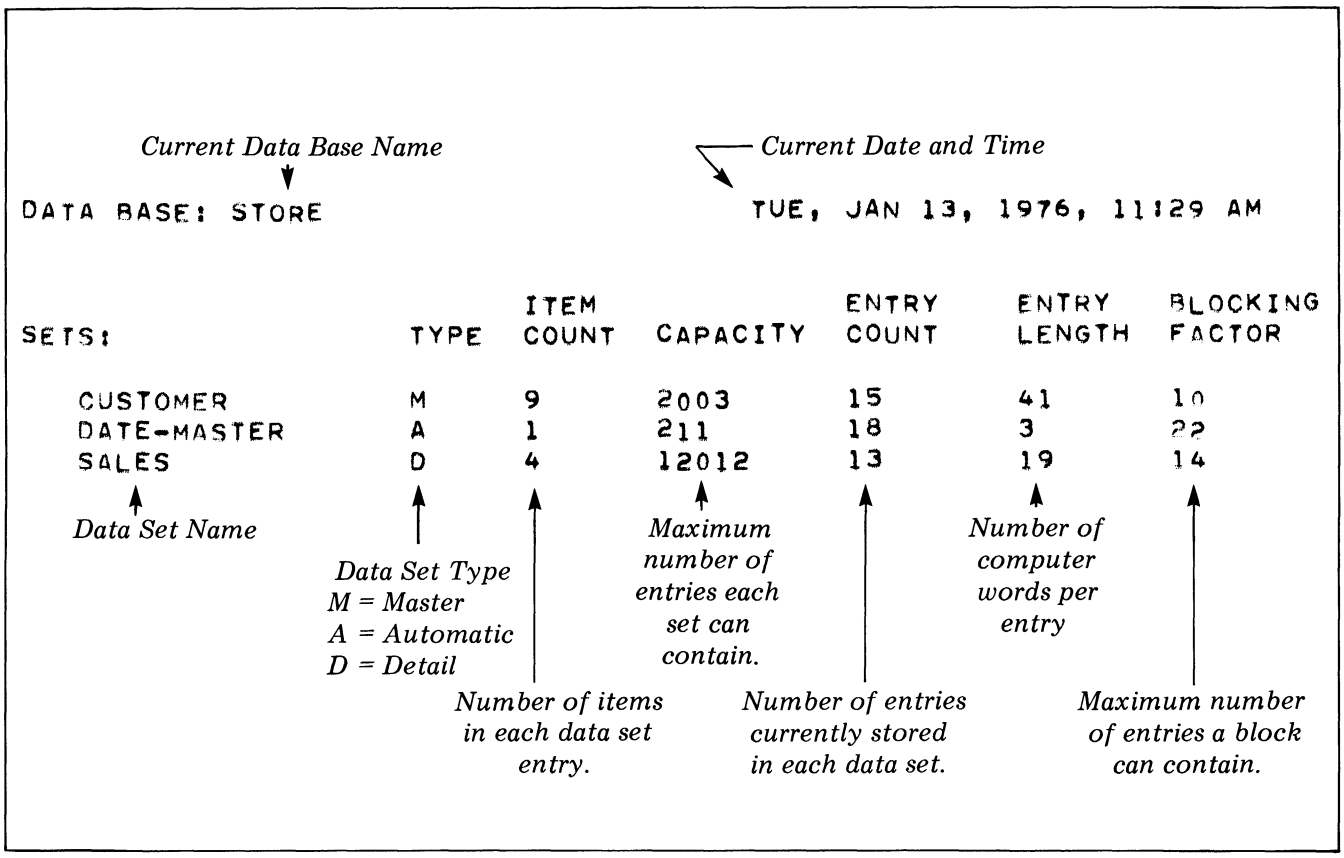


Figure 2-3. FORM SETS Output

```
DATA BASE: STORE                                  TUE, JAN 13, 1976, 11:29 AM

PATH IDENTIFYING INFORMATION

                      ASSOCIATED
MASTER SET NAME       DETAIL SET NAME     SEARCH ITEM NAME     SORT ITEM NAME

CUSTOMER              SALES               ACCOUNT              PURCH-DATE

DATE-MASTER           SALES               PURCH-DATE



                                                               ASSOCIATED
DETAIL SET NAME       SEARCH ITEM NAME    SORT ITEM NAME       MASTER SET NAME

SALES                 ACCOUNT             PURCH-DATE           CUSTOMER
                     !STOCK#
                      PURCH-DATE                               DATE-MASTER
```

Figure 2-4. FORM PATHS Output

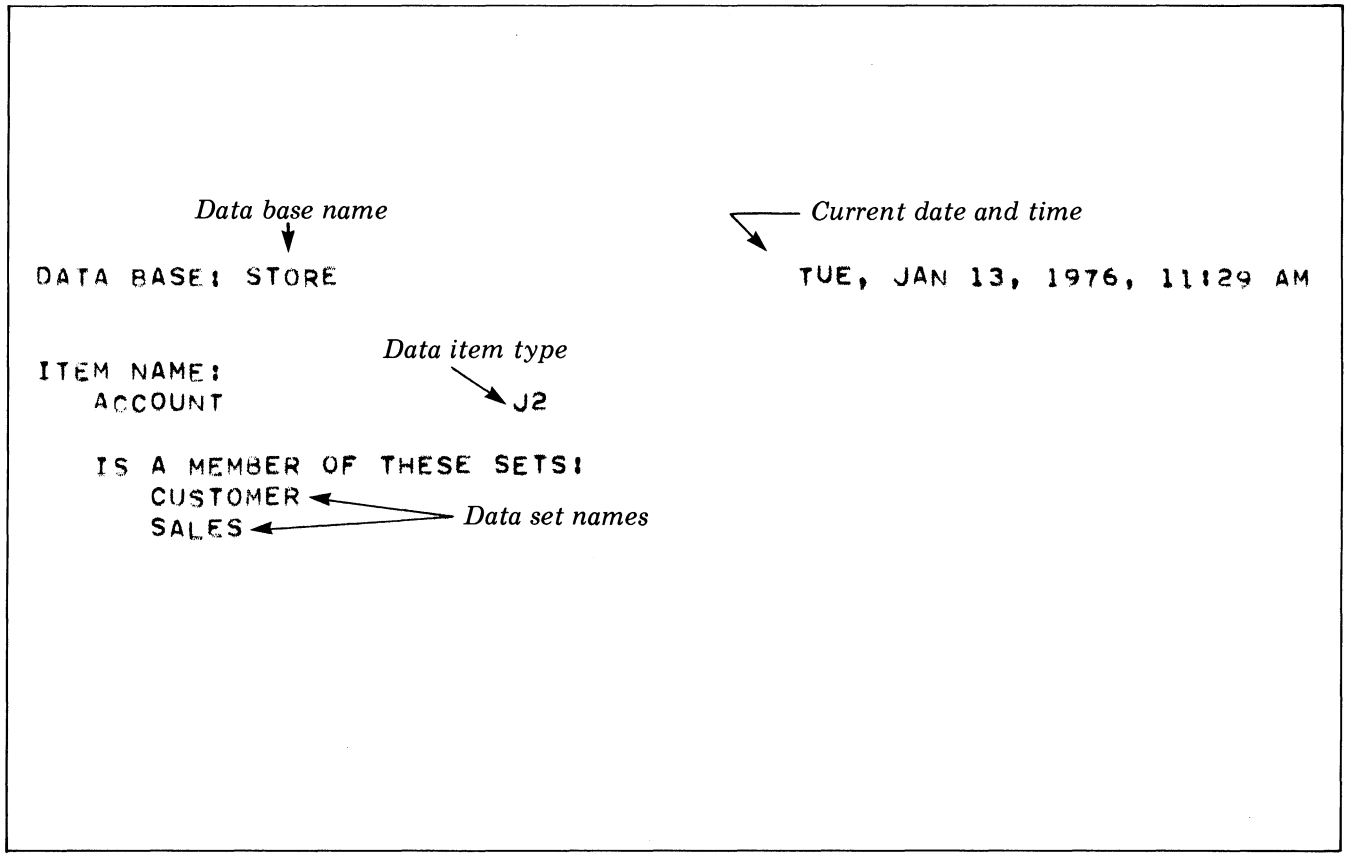


Figure 2-5. FORM *data item name* Output

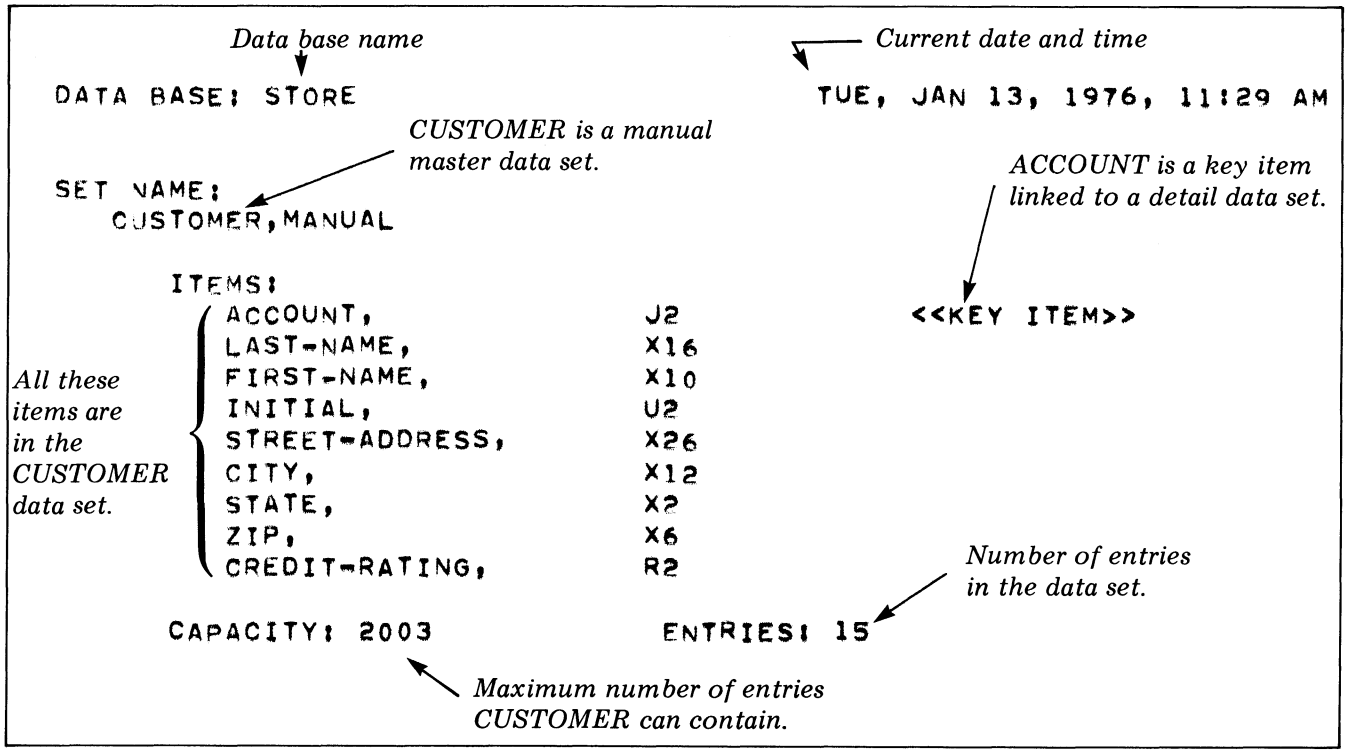


Figure 2-6. FORM *data set name* Output

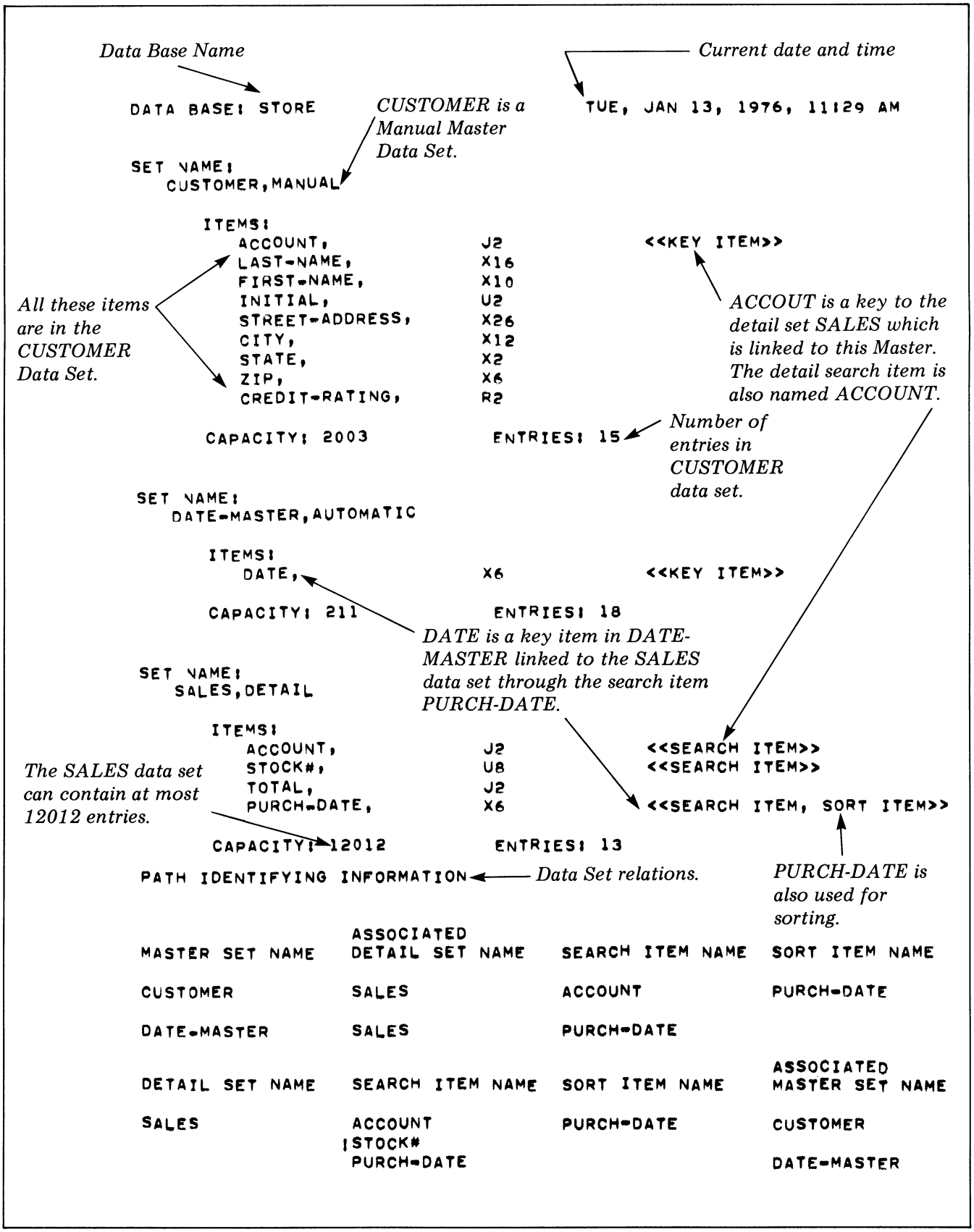


Figure 2-7. FORM Output

HELP

Lists information about the function, format, and parameters of QUERY commands.

The form of the HELP command is

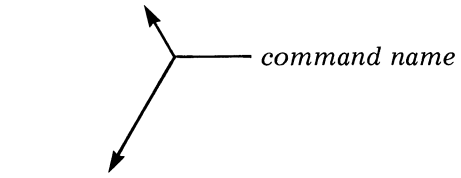
```
{ HELP } [ command name [ FUNCTION ] [ FORMAT ] [ PARAMETERS ] ]
{ H    }                 [ FU       ] [ FO     ] [ PA         ]
```

For example,

HELP

or

HELP FORM FUNCTION



or

HE DEFINE

where	*command name*	consists of any valid QUERY command name such as DISPLAY, EXIT, or RENAME. You may use the short form of the command name (D, E, or REN).
	FUNCTION	indicates that only the command's function should be printed.
	FORMAT	indicates that only the command's format should be printed.
	PARAMETERS	indicates that only the command's parameters should be printed.

The HELP command provides a convenient on-line reference for QUERY commands. If you enter HELP, you receive a list of the QUERY commands followed by a brief description of the function of each command.

If you enter HELP *command name*, you receive information about the specified command's format, function, and parameters, if any.

If you enter HELP *command name* followed by one or more of the parameters FUNCTION, FORMAT, or PARAMETERS, QUERY provides only the information that you request. That is, HELP FIND FUNCTION prints only the FIND command's function and no information about the format or the parameters, and HELP FIND FORMAT PARAMETERS (or HE FI FO PA) prints the format and parameters of the FIND command.

HELP output is listed on the standard list device, unless OUTPUT=LP has been previously entered. In that case, the output is directed to the device equated to the QSLIST file. (See the OUTPUT command description of QSLIST.)

In order to allow you plenty of time to read the command descriptions when using HELP in session mode, QUERY prints the following message after listing several lines:

```
** PRESS ANY KEY TO CONTINUE
```

When you are finished reading the descriptions currently displayed, press any terminal key. The listing will continue.

EXAMPLES

```
>H HELP FUNCTION


FUNCTION -
   TO LIST INFORMATION ABOUT THE FUNCTION, FORMAT
   AND PARAMETERS OF QUERY/3000 COMMANDS
```

You may abbreviate the HELP command and the command you are asking about.

```
>H B


FUNCTION -
   SPECIFY THE DATA BASE TO BE ACCESSED

FORMAT -
   DATA-BASE= DATA BASE NAME

PARAMETERS -
   DATA BASE NAME - NAME OF
< CONTROL Y >

>HELP OUTPUT PA FO
```

You can use Control Y to terminate the command description.

The order of FUNCTION, PARAMETER, AND FORMAT parameters can vary.

```


FORMAT -
   OUTPUT= TERM/LP

PARAMETERS -
   TERM - INDICATES THAT OUTPUT SHOULD BE SENT
          TO THE DEVICE SPECIFIED AS $STDLIST
   LP   - INDICATES THAT OUTPUT SHOULD BE SENT
          TO THE FORMAL FILE DESIGNATOR 'QSLIST'
```

SHOW

Displays the current state of an option that can be enabled or disabled by the ASSIGN command. The only option currently supported by the ASSIGN and SHOW commands is LOCKOPTION.

The form of the SHOW command is

```
{ SHOW } option
{ SH   }
```

For example,

```
SHOW LOCKOPTION
     ↑
option
```

where *option* is the name of an option supported by the ASSIGN command (currently applies only to LOCKOPTION).

Example:

```
>SHOW LOCKOPTION
LOCKOPTION=ON
>ASSIGN LOCKOPTION=OFF
>SHOW LOCKOPTION
LOCKOPTION=OFF
```

VERSION

Displays the current version of QUERY and IMAGE procedures and program files.

The form of the VERSION command is

```
{ VERSION }
{ V       }
```

This command will print the current QUERY version, update, and fix level information and all IMAGE procedures and program files on the standard list device. If OUTPUT=LP has been specified previously, the information is also printed on the QSLIST device. (See the OUTPUT= command for a description of QSLIST.)

EXAMPLE

```
>VERSION

    QUERY       A.03.00

IMAGE PROCEDURES:
    DBOPEN      A.04.00
    DBINFO      A.04.00
    DBCLOSE     A.04.00
    DBFIND      A.04.00
    DBGET       A.04.00
    DBUPDATE    A.04.00
    DBPUT       A.04.00
    DBDELETE    A.04.00
    DBLOCK      A.04.00
    DBUNLOCK    A.04.00
IMAGE PROGRAM FILES:
    DBSCHEMA.PUB.SYS    A.04.00
    DBSTORE.PUB.SYS     A.04.00
    DBRESTOR.PUB.SYS    A.04.00
    DBUNLOAD.PUB.SYS    A.04.00
    DBLOAD.PUB.SYS      A.04.00
    DBUTIL.PUB.SYS      A.04.00
```

LOCATING AND UPDATING DATA

SECTION III

The commands described in this section locate data entries and modify the data base. The FIND command locates data and the UPDATE ADD, UPDATE REPLACE, and UPDATE DELETE commands modify data.

The FIND command locates entries in the data base according to data item values in the entries and provides the following features:

- Up to 50 logical relationships can be specified in one command, and up to 65,000 data entries can be located through one command.
- More than one value can be specified for comparison with a data item.
- The number of entries located (satisfying the logical relationships specified) is reported.
- The FIND command itself can be stored in the Proc-file as a procedure (through the CREATE command) for repeated use without re-entering. FIND procedures can be created which prompt you for the desired search values at procedure execution time, allowing you to search for different values for the same data item each time the procedure is executed.
- Entries may be located from a single data set and in some cases, from multiple data sets.

The format of the FIND command varies with the command's usage. The four possible uses are:

- locating entries in a single data set
- locating all entries in a data set regardless of the value of the data item specified
- locating entries in multiple data sets
- executing a FIND command stored as a procedure.

You may modify your data base in three ways:

- Add data entries to a data set by using UPDATE ADD.
- Delete data entries from a data set by using UPDATE DELETE.
- Modify data entries in a data set by changing the values of data items not defined as search or sort items in the data base by using UPDATE REPLACE.

UPDATE DELETE and UPDATE REPLACE are extensions of the FIND command, in that they operate on the entries selected by the last FIND command executed. UPDATE ADD does not require a previous FIND command.

UPDATE commands can be stored as procedures in a Proc-file and used at a later time. The procedure must reside in the currently defined Proc-file.

In order to use the UPDATE commands your mode must be less than 5. If you open the data base with MODE equal to 2, the only UPDATE command you may use is REPLACE. With mode 1, 3, or 4 you may use all UPDATE commands.

USING THE FIND COMMAND

The FIND command consists of one or more logical relations consisting of a data item name, relational operator, and one or more values separated by commas. QUERY searches the appropriate data set(s) and stores the relative record addresses (in an internal buffer not directly accessible to you) of the data entries which satisfy the relations. The number of entries located is reported with the message:

xxx ENTRIES QUALIFIED

After the command is entered, QUERY may type the message:

USING SERIAL READ

This means that the data set must be searched in serial fashion without the benefit of master data set indexing. Serial searching can (in some cases) consume a great deal of time. You may abort the search by pressing Control Y if you decide you do not want to wait for its completion.

Once data entries have been located they are available for listing (using the REPORT command), updating (using the UPDATE REPLACE command), or deleting (using UPDATE DELETE). The located entries remain available at any time until QUERY execution terminates, the entries are deleted, or another FIND command is entered. This means, for example, that located entries can be listed using one report format and then again using another format without using FIND again.

Locates entries in a single data set.

The form of the FIND command when used for this purpose is

{FIND / F} *relation* [{AND / OR} *relation* ...] [END]

For example

FIND BADGE# IS "09.18" END
relation

or

FIND L-NAME IS CLEVER AND F-NAME IS JACK END
relation *relation*

or

FI STATE IS "CALIFORNIA", "NEVADA", "WASHINGTON" END
relation

or

FIND EMPLOYEE.SALARY IGT 1000 END
data set name *data item name* *relop* *value*

where

relation	takes the form: *[data set name.] data item name relop "value" [,"value". .]*
data set name	is the name of a data set in the current data base.
data item name	is the name of a data item contained in the data base. If a *data set name* is used, the data item must belong to the specified data set.
relop	is a relational operator as shown in table 3-1.
value	is the data item value. It must be the same type and within the same value range as the data item named in the *relation*. *Value* need not be enclosed in quote marks (") unless the value contains special characters. *Values* which are not contained in quotes are upshifted. For example, California is converted to CALIFORNIA before it is compared to data item values in the data base.

Table 3-1. FIND Command Relational Operators

OPERATOR	MEANING
= IS IE EQ	is equal to
# ISNOT INE NE <>	is not equal to
ILT LT <	is less than
INLT GE >=	is not less than (is greater than or equal to)
IGT GT >	is greater than
INGT LE <=	is not greater than (is less than or equal to)
IB *value* $_1$,*value*$_2$	is between (and including) *value*$_1$ and *value*$_2$

multiple values may be used

The >=, <=, <> operators cannot have any intervening spaces.

U AND X TYPE VALUES

When entering values for U and X type data items, the values must appear exactly as the data was entered originally. For example, if the data item STREET-ADDRESS has a value with three spaces between the street number and name, you must enter those spaces or QUERY will not find the item. Leading blanks must also be entered if they appear in the item. Since blanks or spaces are special characters, all such values must be enclosed in quotes.

LOGICAL CONNECTORS

To make more than one comparison for each entry selected, you connect relations with the logical connector AND or OR. The AND connector instructs QUERY to select only those entries whose data item values satisfy both relations on either side of the AND. The OR connector indicates that entries are selected if one or the other (or both) of the two relationships on either side of the OR is satisfied.

Both types of logical connectors can appear in a FIND command. All relations connected with AND are examined as if they were surrounded by parentheses. Any relation or set of relations separated from others by OR are compared to the data entry and the entry is selected if the relations are true. For example, the command:

```
FIND  A = 3 AND B = 4  OR C IGT 9  END
```

locates all entries with both A equal to 3 and B equal to 4 as well as all entries containing a C value greater than 9. The command:

```
FIND  A = 3 OR A = 2  AND B = 5  AND C = 8  OR  B = 9  END
```

locates all entries with either A equal to 3, A equal to 2 and B equal to 5 *and* C equal to 8, or entries with B equal to 9.

Parentheses cannot be used in a command, but you can create constructs which act as parentheses and force an OR comparison to take precedence over an AND comparison. For example, if C*n* stands for a relation,

$(C_1$ OR $C_2)$ AND C_3

is represented as:

C_1 AND C_3 OR C_2 AND C_3

Up to 50 logical connectors may be used in one FIND command.

MULTIPLE VALUES

To specify more than one value for the same data item, list the values one after the other, separated by commas. For example, the command:

```
FIND STATE IS "CALIFORNIA", "NEVADA", "WASHINGTON" END
```

locates entries with the value of data item STATE equal to either CALIFORNIA,NEVADA, or WASHINGTON. The above command is equivalent to:

```
FIND STATE IS "CALIFORNIA" OR STATE IS "NEVADA" OR STATE IS "WASHINGTON" END
```

Multiple values may be used only with the "equal" or "not equal" relational operators.

FIND

USING NULL VALUES

A FIND command can be entered which prompts you for data item values to be compared with data item values in data entries of the data set. To do this, you merely use null data item values in the command. Null values are represented by a pair of quote marks without any intervening characters or blank spaces (""). When the command is executed, you are prompted to enter a value for each null value in the command. This is useful when the FIND command is stored as a procedure on the Proc-file. (See Section V for more information about procedures.)

The procedure can be executed using different comparison values without modifying the procedure each time. For example, the command:

```
>FIND SALES.ACCOUNT # "" END
```

would prompt you for a value of ACCOUNT with the message:

```
WHAT IS THE VALUE OF - ACCOUNT
>>24536173
USING SERIAL READ
10  ENTRIES QUALIFIED
```

QUERY searches the appropriate data set for the value you specify. The value should be entered *without* surrounding quotes since all characters entered (including leading blanks, quotes, and other special characters) are significant. The maximum value size is 72 characters.

You are prompted once for each null value in the command. For example, the command:

```
>FIND CUSTOMER.CITY IS "","" END
```

prompts you twice for the value of CITY as follows:

```
WHAT IS THE VALUE OF - CITY
>>PETALUMA
WHAT IS THE VALUE OF - CITY
>>ALVISO
USING SERIAL READ
2  ENTRIES QUALIFIED
```

Lowercase characters are upshifted unless the data item type is X.

In session mode, supply the desired value after each prompt message. In job mode, you must anticipate the order of prompts and supply the desired values, one per record, following the FIND command.

EXAMPLES

```
>FIND LAST-NAME IS MARTENSEN END
USING SERIAL READ
Ø  ENTRIES QUALIFIED
```

To determine whether or not a customer is already in the data set, you can try to find their name.

```
>FIND CUSTOMER.ACCOUNT EQ 24536173
1  ENTRIES QUALIFIED
>S=
>FIND ACCOUNT EQ 24536173
ACCOUNT            IS A MEMBER OF THESE SETS:
CUSTOMER, SALES
WHICH SET DO YOU WISH TO USE?
>>SALES
3  ENTRIES QUALIFIED
```

Since account is in more than one data set you can either qualify the data item name or let QUERY prompt for it.

SALES is automatically entered in the data set list.

In the example below, QUERY uses the data set list to determine which of the three data sets containing STOCK# to use. Since SALES was placed in the data set list in the previous example, it is used. However, the data item DESCRIPTION is in the PRODUCT data set and a FIND command can only refer to one data set so an error results. It is usually best to clear the data set list if you are unsure about what it contains.

```
>F STOCK# IGT 33333333 AND DESCRIPTION IS "NEHRU JACKET"
RETRIEVAL FROM MORE THAN ONE DATA SET
>S=
>F STOCK# IGT 33333333 AND DESCRIPTION IS "NEHRU JACKET"
STOCK#             IS A MEMBER OF THESE SETS:
PRODUCT, SALES, INVENTORY
WHICH SET DO YOU WISH TO USE?
>>PRODUCT
USING SERIAL READ
Ø  ENTRIES QUALIFIED
>F STOCK# IGT 33333333 AND DESCRIPTION IS "NEHRU JACKET"
USING SERIAL READ
Ø  ENTRIES QUALIFIED
```

Clear the data set list.

Now QUERY prompts for the data set.

PRODUCT was entered in the data set list automatically as a result of the previous command so QUERY uses PRODUCT and does not prompt for the data set name.

```
>F ACCOUNT IGT 55555555 AND STATE IS CA OR ACCOUNT IS 12121212&
>>AND STATE IS MA OR CUSTOMER.STATE IS AZ
ACCOUNT            IS A MEMBER OF THESE SETS:
CUSTOMER, SALES
WHICH SET DO YOU WISH TO USE?
>>CUSTOMER
INVALID NUMERIC DIGIT
```

This error occurs because an ampersand is used to continue the line but there is no space before it or at the beginning of the next line.

```
>F CUSTOMER.ACCOUNT IGT 55555555 AND STATE IS CA OR ACCOUNT &
>> IS 121212 AND STATE IS MA OR STATE IS AZ
STATE              IS A MEMBER OF THESE SETS:
CUSTOMER, SUP-MASTER
WHICH SET DO YOU WISH TO USE?
>>CUSTOMER
USING SERIAL READ
6  ENTRIES QUALIFIED
>R D, ACCOUNT, 10; D, STATE, 15; END
```

All items which appear in multiple sets must be qualified unless the set name is in the set list.

After the entries are located you can use the REPORT command to print the data

```
  76623455    CA
  74001813    CA
  87654321    CA
  80808080    CA
  77765555    CA
  99998877    CA
```

```
>F CUSTOMER.ACCOUNT IGT 55555555 AND CUSTOMER.STATE IS CA OR &
>>ACCOUNT IS 12121212 AND STATE IS MA OR STATE IS AZ
USING SERIAL READ
6  ENTRIES QUALIFIED
```

STATE only needs to be qualified once.

A simpler technique for determining the data set to be used is to enter it in the data set list with the DATA-SETS= command

```
>S=CUSTOMER
>F ACCOUNT IGT 55555555 AND STATE IS CA OR ACCOUNT IS 12121212 &
>>AND STATE IS MA OR STATE IS AZ
USING SERIAL READ
6  ENTRIES QUALIFIED
>
```

FIND ALL

Locates all entries in the data set regardless of the value of the data item specified.

> The form of the FIND command when used for this purpose is
>
> $\left\{\begin{matrix}\text{FIND}\\ \text{F}\end{matrix}\right\}$ ALL *[data set name.] data item name*

For example,

FIND ALL LABOR.BADGE#

data set name *data item name*

or

FI ALL F-NAME

data item name

where *data set name* is the name of the data set you want to access.

data item name is the name of any item in that set.

If you want to locate all entries in a data set (and use these entries in a report, for example), this is an easy way to do so.

If you do not specify a *data set name*, QUERY will check the data set list and follow the same rules as defined when using a FIND command with a single data set. (See the first form of the FIND command earlier in this section.)

EXAMPLES

```
>F ALL CUSTOMER.ACCOUNT
USING SERIAL READ
13  ENTRIES QUALIFIED
```

All entries in the CUSTOMER data set are located.

FIND procedure

Executes a FIND procedure stored in the current Proc-file.

The form of the FIND command when used for this purpose is

$$\left\{\begin{matrix}\text{FIND}\\ \text{F}\end{matrix}\right\} \quad \textit{procedure name [,character]}$$

For example,

FIND PROCA

procedure name

or

FI USERS,X

procedure name *character*

where *procedure name* is the name of a FIND command previously stored as a procedure using the CREATE command. The procedure must exist in the current Proc-file specified with the PROC-FILE= command or in response to the PROC-FILE prompt.

character is any printable ASCII character. If character is included in the command, then the FIND procedure is listed.

QUERY searches the current Proc-file (defined by a DEFINE or PROC-FILE) and executes the procedure named in the command. If the Proc-file has not been declared, or the procedure does not exist in the Proc-file, or the procedure is incorrect in some way, you are informed by

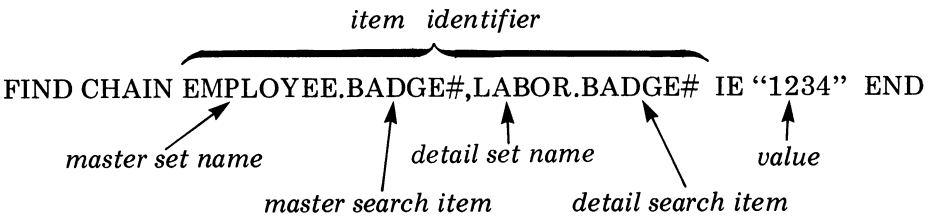
FIND CHAIN

Locates data entries from more than one data set.

The form of the FIND command when used for this purpose is

$$\left\{\begin{matrix}\text{FIND}\\ \text{F}\end{matrix}\right\}\ \text{CHAIN}\ \textit{item identifier}\left\{\begin{matrix}\text{IS}\\ \text{IE}\\ \text{EQ}\\ =\end{matrix}\right\}\text{“}\textit{value}\text{”}\left[\left\{\begin{matrix}\text{AND}\\ \text{OR}\end{matrix}\right\}\ \textit{item identifier}\left\{\begin{matrix}\text{IS}\\ \text{IE}\\ \text{EQ}\\ =\end{matrix}\right\}\ \text{“}\textit{value}\text{”}\ \ldots\right]\left[\text{END}\right]$$

For example,



or

```
FI CHAIN EMPLOYEE.BADGE#,LABOR.BADGE# IS 1234 OR &

EMPLOYEE.BADGE#,LABOR.BADGE# IS 9018 END
```

indicates command continues in next record

where

item identifier	takes the form: *master set name.master search item,detail set name.detail search item*
master set name	is the name of a master data set in the data base.
master search item	is the name of a data item defined as the search item for the master data set.
detail set name	is the name of a detail data set that is related to the master data set previously named.
detail search item	is the name of a data item defined as a search item for the detail data set. This search item must provide the link between the detail set and the previously named master data set.
value	is a data item value. It must be the same type and within the same value range as the data items *master search item* and *detail search item*. *Value* must be enclosed in quotes only if it contains special characters or blanks.

Only one detail data set can be accessed using a FIND CHAIN command but multiple master sets may be used if they relate to the same detail set through search items.

FIND CHAIN

The data base administrator can provide you with the information required to use this command or you can use the FORM command to determine the data item and data set relations.

When you enter FIND CHAIN, QUERY searches the specified master data sets for an entry containing the specified search item value. Then QUERY searches the specified detail data set for entries containing the same search item value. Detail entries with the same search item value are called *chains*. The effect of FIND CHAIN is to locate all the members of a detail chain and the master data entry which constitutes the chain head. For more information about chains, chain heads, and data set relations, consult the *IMAGE/3000 Reference Manual* (see Preface for part number).

Remember, when using FIND CHAIN:

- Only selection on the basis of equality can be made. Relational operators other than IS, IE, EQ, and = are not allowed.
- Only one value per relational operator is allowed.
- Data items named in the command must always be search items (as defined in the data base schema).
- Multiple logical connectors (AND, OR) can appear in the same command. The rules for determining which entries are described with the FIND command.

EXAMPLES

```
>FIND CHAIN CUSTOMER.ACCOUNT,SALES.ACCOUNT IS 76623455,24536173 &
>>AND PRODUCT.STOCK#,SALES.STOCK# IGT 22222222 OR &
>>CUSTOMER.ACCOUNT,SALES.ACCOUNT ILT 55555555 END
INVALID # OF VALUES FOR RELATIONAL OPERATOR
```

Only one value is allowed per relation.

```
>FIND CHAIN CUSTOMER.ACCOUNT,SALES.ACCOUNT IS 76623455 AND &
>>PRODUCT.STOCK#,SALES.STOCK# IGT 22222222 OR &
>>CUSTOMER.ACCOUNT,SALES.ACCOUNT ILT 55555555 END
INVALID RELATIONAL OPERATOR
```

Only "equal" relational operators are allowed.

In the example below, CUSTOMER is searched for an entry containing ACCOUNT equal to 76623455 and one entry is found. Then PRODUCT is searched for an entry containing STOCK# equal to 6550D22S, one entry is found. SALES is searched for entries with both ACCOUNT equal to 76623455 and STOCK# equal to 6550D22S but no entries are located. CUSTOMER is searched for entries containing ACCOUNT equal to 54283545 and SALES is searched for entries containing ACCOUNT equal to 54283545. One entry is found for each relation.

```
>FIND CHAIN CUSTOMER.ACCOUNT,SALES.ACCOUNT IS 76623455 AND &
>>PRODUCT.STOCK#,SALES.STOCK# IS 6550D22S OR &
>>CUSTOMER.ACCOUNT,SALES.ACCOUNT IS 54283545 END
4  ENTRIES QUALIFIED
>R ALL
```

Four entries were located which satisfied the relations. The REPORT ALL command prints the content of the entries.

```
ACCOUNT           =76623455
LAST-NAME         =MCFALL
FIRST-NAME        =JEFFREY
INITIAL           =X
STREET-ADDRESS    =6650 MONTEREY ROAD
CITY              =CARMEL
STATE             =CA
ZIP               =93921
CREDIT-RATING     = 3.20000
```

CUSTOMER data set entry. (Satisfies CUSTOMER.ACCOUNT IS 76623455).

```
STOCK#            =6550D22S
DESCRIPTION       =BASEBALL BAT
```

PRODUCT data set entry. (Satisfies PRODUCT. STOCK# IS 6550D22S).

```
ACCOUNT           =54283545
LAST-NAME         =MAYFIELD
FIRST-NAME        =WILLIAM
INITIAL           =
STREET-ADDRESS    =39 41ST AVE.
CITY              =PETALUMA
STATE             =GA
ZIP               =10101
CREDIT-RATING     = 8.50000
```

CUSTOMER data set entry. (Satisfies CUSTOMER.ACCOUNT IS 54283545).

```
ACCOUNT           =54283545
STOCK#            =4397D13P
QUANTITY          =1
PRICE             =4590
TAX               =276
TOTAL             =0
PURCH-DATE        =121575
DELIV-DATE        =122075
```

SALES data set entry. (Satisfies SALES.ACCOUNT IS 54283545).

```
>
```

Note that ACCOUNT and STOCK# are both search items.

UPDATE ADD

Adds data entries to a manual master or detail data set.

The form of the UPDATE ADD command is

```
{UPDATE  ADD,}
{U ADD,      }
{ADD  [,]    }   data set name
{AD   [,]    }
```

For example,

```
UPDATE  ADD,  LABOR
                ↖
              data set name
```

or

```
AD,  PAYROLL
       ↑
   data set name
```

where *data set name* is the name of a data set in the data base. The data set must be either a manual master or detail, and you must have write access as determined by your response to the PASSWORD=>> prompt when the data base was last specified or use of the PASSWORD= command.

When you use the UPDATE ADD command, QUERY prompts you for data item values by printing the names of the data items. You type the desired value following the data item name. For example,

```
>AD PRODUCT
STOCK#          =>>12345678
DESCRIPTION     =>>"ANTIMACASSAR"

STOCK#          =>> return
STOCK#          =>>54231234
DESCRIPTION     =>> return

STOCK#          =>>//
>
```

Values must be provided for search items.

Two slashes terminate the command.

If a value contains invalid characters, QUERY issues a message and reprompts for the same data item name. Character type values are truncated if too long.

USING QUOTE MARKS

Values may be entered with or without bracketing quote marks. When you enter a U or X type value without quotes, leading blanks are ignored and the value is left-justified in the data item field. Blanks or any other special characters are entered in the data field only if the value is entered with quotes. For example,

```
F-NAME>>  SALLY
```

is entered in an 8-character field as: and

```
F-NAME>>"  SALLY"
```

is entered in the same field as:

A U type data item is automatically upshifted if entered in lower case. You can override the upshifting by enclosing the entry in quotes. For example, if F-NAME is a U type data value,

```
F-NAME>>sally
```
is entered as **SALLY**

but,

```
F-NAME>>"sally"
```
is entered as **sally**

If quote marks are to be included as part of the value, they must be entered twice to avoid confusion as delimiters. For example, the value:

ROBERT "BOB" BRUN

must be entered as:

```
"ROBERT ""BOB"" BRUN"
```

The value must also be surrounded by quotes in this case.

Numeric type values (all types except U and X) are always right-justified whether entered with or without quotes. Numbers entered in X type data items are left-justified just as other character type values.

NULL VALUES

Only search and sort items must have values supplied. If you do not want to enter a value for other items you may enter a null value. In session mode, press *return* without any preceding characters. In job mode supply a blank record to indicate a null value. Numeric data items which do not contain a value are set to zero, character items are set to blanks.

REAL NUMBER VALUES

Real number values can be entered either as fixed or floating point numbers. An example of a fixed-point number is:

```
23.45
```

while an example of a floating-point number is:

```
2.345E+01
```

The signed integer following the E stands for a power of 10 to be multiplied by the number to the left of the E. Both examples above stand for the same number.

UPDATE ADD

DETAIL DATA SETS

If a detail data set is linked to one or more *manual* master data sets, you must know the content of the master data set before you can add entries to the detail data set. Each detail entry is associated with a master entry through a search item. The detail search item and the master search item must have the same value and the value must exist in a master entry before you can add it to a detail entry. If you enter a search item value that does not exist in the master data set linked to the detail through that item, QUERY prints the message:

MISSING CHAIN HEAD FOR *data item name / data set name*

and does not place the entry in the data set. For example:

```
>AD INVENTORY <-------------------------------------- detail set
STOCK#          =>>9999A99A
ONHANDQTY       =>>287
SUPPLIER        =>>XXXXXX <----------------------------- search item
UNIT-COST       =>>1455
LASTSHIPDATE    =>>121875
BINNUM          =>>9
MISSING CHAIN HEAD FOR SUPPLIER/SUP-MASTER
```

There is no SUPPLIER value XXXXXX in the SUP-MASTER data set. You must add an entry to SUP-MASTER giving XXXXXX as the value for the data item named SUPPLIER. Then you will be allowed to add the same value to the detail set named INVENTORY.

TERMINATING UPDATE ADD

Once you have supplied values (null or otherwise) for all the data items in the entry, QUERY begins prompting you for another entry. To continue adding entries, merely enter an appropriate value for the prompt. To terminate the UPDATE ADD command, either enter two slashes (/ /) followed by *return* or enter Control Y (Y^C). (Control Y is entered by pressing and holding the *control* key and then striking Y.)

The command can be terminated at any time you are prompted for a value. However, unless you want to discard the values you have entered for the current entry, you should not use Y^C or two slashes before you have been prompted for the last value of the current entry. In other words, you normally terminate the command in response to the first data item prompt in an entry.

EXAMPLES

```
>UPDATE ADD,PRODUCT
STOCK#              =>>9898989898
INPUT TOO LONG - TRUNCATED
DESCRIPTION         =>> return

STOCK#              =>>78787878
DESCRIPTION         =>>SQUIRREL CAGE

STOCK#              =>>//
>UPDATE ADD,SALES
ACCOUNT             =>>80808080
STOCK#              =>>78787878
QUANTITY            =>>6
PRICE               =>>2000
TAX                 =>> return
TOTAL               =>> return
PURCH-DATE          =>>121575
DELIV-DATE          =>>121675
MISSING CHAIN HEAD FOR ACCOUNT/CUSTOMER

ACCOUNT             =>>//
>UPDATE ADD,CUSTOMER
ACCOUNT             =>>80808080
LAST-NAME           =>>CELERY
FIRST-NAME          =>>ALLISON
INITIAL             =>>B.
STREET-ADDRESS      =>>18 ASCOT AVE.
CITY                =>>CARMEL
STATE               =>>CA
ZIP                 =>>93921
CREDIT-RATING       =>>1.2

ACCOUNT             =>>//
>UPDATE ADD,SALES
ACCOUNT             =>>80808080
STOCK#              =>>78787878
QUANTITY            =>>6
PRICE               =>>2000
TAX                 =>> return
TOTAL               =>> return
PURCH-DATE          =>>121575
DELIV-DATE          =>>121675

ACCOUNT             =>>//
>
```

You need not enter a value if the item is not a search item.

Terminate command with / /.

An attempt is made to enter an account which is not in the CUSTOMER master data set.

QUERY prints an error

Terminate the command and add the account to the master data set.

Terminate the command and now add the original entry to the SALES detail set.

Note: These dates are automatically added to the automatic master data set named DATE-MASTER if they are not already in it.

UPDATE DELETE

Deletes data entries from a data set.

The form of the UPDATE DELETE command is

{ UPDATE DELETE
  U DELETE
  DELETE
  DEL }

For example,

UPDATE DELETE

or

DELETE

This command is an extension of the FIND command in that it deletes those entries selected by the previous FIND. All of the entries must reside in the same data set. Entering FIND CHAIN (selecting entries from more than one data set) and then UPDATE DELETE is not allowed.

SECURITY PROVISIONS

To delete entries from a data set, you must possess write access to the data set, as determined by the password you enter in response to the latest PASSWORD= prompt or through the latest use of the PASSWORD= command.

When you enter an UPDATE DELETE command, QUERY prints the message:

```
DELETE ALL RETRIEVED ENTRIES (YES OR NO)?
>>
```

The message reminds you that all the entries selected by the last FIND will be deleted by the command. If you respond YES, QUERY deletes the entries. If you respond NO, the command is ignored and you are prompted for another command. The message is not printed in job mode.

MASTER DATA SET ENTRIES

QUERY disallows any attempt to delete the master entry if its search item value still exists in the search items of the appropriately linked detail data sets.

EXAMPLES

```
>F CUSTOMER.ACCOUNT=88888888,11111111
2  ENTRIES QUALIFIED
>DEL
DELETE ALL RETRIEVED ENTRIES (YES OR NO)?
>>YES
>
```

The CUSTOMER entries with ACCOUNT equal to 88888888 or 11111111 are deleted.

```
>F CUSTOMER.ACCOUNT IS 80808080
1  ENTRIES QUALIFIED
>UPDATE DELETE
DELETE ALL RETRIEVED ENTRIES (YES OR NO)?
>>YES
ATTEMPTED DELETION OF CHAIN HEAD
```

In this example, there is still a detail data entry in SALES with an ACCOUNT value of 80808080 so QUERY will not delete the entry from the CUSTOMER master data set.

```
>F CHAIN CUSTOMER.ACCOUNT,SALES.ACCOUNT IS 80808080 END
2  ENTRIES QUALIFIED
>UPDATE DELETE
UPDATE TO MORE THAN ONE SET
>FIND SALES.ACCOUNT IS 80808080 END
1  ENTRIES QUALIFIED
>UPDATE DELETE
DELETE ALL RETRIEVED ENTRIES (YES OR NO)?
>>YES
>
```

FIND CHAIN is not allowed for deleting entries.

An entry with ACCOUNT equal to 80808080 is deleted from the SALES detail data set.

UPDATE REPLACE

Modifies the values of data items.

The form of the UPDATE REPLACE command is

{ UPDATE REPLACE
U REPLACE
REPLACE
REPL } *,data item name = "value"; [data item name="value";. . .] END*

For example,

```
UPDATE REPLACE, DATE="741010"; END
```

data item name *value*

or

```
REPLACE,
DATE="741010";
HOURS="4";
END
```

where *data item name* is the name of the data item contained in the entries selected by the last FIND command.

value is the new value for the data item surrounded by quotes. It must be the proper size and type for the named item. Quotes are required. Type U, X, Z, and P values are truncated if too large. QUERY prints an error for other values of improper size or type.

UPDATE REPLACE is an extension of the FIND command in that it operates on all data entries selected by the previous FIND command. All of the data entries selected must be from the same set. Entering a FIND CHAIN command (which selects entries from a detail and one or more master data sets) followed by an UPDATE REPLACE command is not allowed.

QUERY does not allow you to modify data items defined in the data base as search or sort items.

The UPDATE REPLACE command consists of a series of data item name/data item value pairs separated by semicolons. When the command is entered, QUERY replaces the value of each data item named in the command with the new value enclosed in quotes. The names of data items appearing in the command must belong to the data entries selected by the last FIND command.

SESSION MODE

When you enter an UPDATE REPLACE command from a terminal, QUERY checks each line of the command as it is entered. If an error occurs, the line is ignored, an error message is sent, and you are prompted for another line. QUERY continues to prompt for lines until you enter END. END is required.

The replacement statements (data item name/value pairs) may be entered on one line:

```
>UPDATE REPLACE, DATE="741010";HOURS="4";END
```

or on several lines:

```
>UPDATE REPLACE,
>>DATE="741010";
>>HOURS="4";
>>END
>
```

Note: An & is not required here to continue the command in the next record.

JOB MODE

If you are entering the command in job mode, the replacement statements may appear in one record or several consecutive records.

NULL VALUES

You may store an UPDATE REPLACE command as a procedure with null values for the data items. You do this by entering a pair of quotation marks with no intervening characters after *data item name=* (for example, DATE=" ").

When such a procedure is executed using the UPDATE *procedure name* command, QUERY prompts you for a value by printing the message:

```
WHAT IS THE VALUE OF data item name
>>
```

where *data item name* is the name of the data item associated with the null value in the UPDATE REPLACE command. The desired value must be entered without quotes. All characters entered (including blanks, quotes, and other special characters) are significant. The maximum number of characters which may be entered as a value is 72. Lowercase characters are upshifted unless the data item is type X.

UPDATE REPLACE

EXAMPLES

```
>F CUSTOMER.FIRST-NAME=GENEVA
USING SERIAL READ
1  ENTRIES QUALIFIED
>REPL,STREET-ADDRESS="868 DOYLE ROAD"
>>END
>UPDATE REPLACE,FIRST-NAME="GINGER";INITIAL="K";END
>LIST CUSTOMER FOR FIRST-NAME=GENEVA
```

The CUSTOMER entry with FIRST-NAME of GENEVA is located. The street address is changed.

Then the first name and initial are replaced.

```
>LIST CUSTOMER FOR FIRST-NAME=GINGER
```

The LIST command can be used to check the results.

```
   ACCOUNT   LAST-NAME          FIRST-NAME   IN   STREET-ADDRESS

  10034765   SLATER             GINGER       K    868 DOYLE ROAD
```

```
>AD SALES
ACCOUNT          =>>80808080
STOCK#           =>>78787878
QUANTITY         =>>6
PRICE            =>>2000
TAX              =>>return
TOTAL            =>>return
PURCH-DATE       =>>121575
DELIV-DATE       =>>121675
```

No values are entered for TAX and TOTAL when this entry is added.

```
ACCOUNT          =>>//
>UPDATE REPLACE,TAX=25;TOTAL=2025;END
RECORD HAS NOT YET BEEN FOUND
>F SALES.ACCOUNT IS 80808080
1  ENTRIES QUALIFIED
>UPDATE REPLACE,TAX="25";TOTAL="2025";END
```

You can use the UPDATE REPLACE command to add these values later . . . but first you must locate the entry.with the FIND command

OR

```
>UPDATE REPLACE,
>>TAX="25"
>>TOTAL="2025"
>>END
```

EXAMPLES

```
>FIND PRODUCT.STOCK# IGT 70000000
USING SERIAL READ
3  ENTRIES QUALIFIED
>UPDATE REPLACE,
>>DESCRIPTION=OBSOLETE
EXPECTED A LITERAL VALUE
>>DESCRIPTION="OBSOLETE"
>>END
```

Values must be enclosed in quotes.

```
>REPORT ALL

STOCK#              =7391Z22F
DESCRIPTION         =OBSOLETE

STOCK#              =78787878
DESCRIPTION         =OBSOLETE

STOCK#              =98989898
DESCRIPTION         =OBSOLETE
```

You can use REPORT ALL to verify the replacements. The entries are still available from the last FIND. Each PRODUCT data set entry with STOCK# equal to 70000000 is changed so that the description is OBSOLETE.

UPDATE procedure

Executes an UPDATE command stored as a procedure in the current Proc-file.

The form of the UPDATE procedure command is

{UPDATE / U} *procedure name [,character]*

For example,

UPDATE PROCX
(PROCX: *procedure name*)

or

UP CHANGE,%
(CHANGE: *procedure name*; %: *character*)

where *procedure name* is the name of an UPDATE ADD, UPDATE REPLACE, or UPDATE DELETE command procedure in the current Proc-file.

character is any printing ASCII character.

You can use the CREATE command to store any of the three forms of the UPDATE command. Once the procedure is stored in the Proc-file, you use this form of the UPDATE command to execute it. Instructions for creating a command procedure are given in Section V. The UPDATE ADD, UPDATE REPLACE, and UPDATE DELETE command forms are described in this section.

If you use an UPDATE procedure using the UPDATE *procedure name* form of the command, QUERY first checks each line of the procedure. If an error occurs, a message is sent and the incorrect statement in the line is ignored. All other correct statements in the procedure are executed. If the *character* is included in the command, the procedure is listed before execution begins.

EXAMPLES

```
>F LAST-NAME IS MAYFIELD
USING SERIAL READ
1  ENTRIES QUALIFIED
>PROC-FILE=PROCX
>UPDATE CHANGE
WHAT IS THE VALUE OF CITY
>>PETALUMA
WHAT IS THE VALUE OF STREET-ADDRESS
>>37 41ST AVE.
WHAT IS THE VALUE OF CREDIT-RATING
>>8.5
```

Locate the entry to be updated.

The procedure named CHANGE (located in Proc-file PROCX) contains:

U REPLACE,
CITY=" ";
STREET-ADDRESS=" ";
CREDIT-RATING=" ";
END

QUERY prompts for values when the procedure is executed.

```
>FIND F101
WHAT IS THE VALUE OF - ACCOUNT
>>11111111
NO ENTRY

NO ENTRY

0  ENTRIES QUALIFIED
>FIND F101
WHAT IS THE VALUE OF - ACCOUNT
>>54283545
2  ENTRIES QUALIFIED
>R ALL,Z
```

The F101 procedure contains:

```
FIND CHAIN
CUSTOMER.ACCOUNT,SALES.
   ACCOUNT
IS " " END
```

QUERY prompts for the value of ACCOUNT.

The REPORT ALL command prints the entry data.

CUSTOMER master data set entry.

```
54283545
MAYFIELD
WILLIAM

37 41ST AVE.
PETALUMA
GA
10101
 8.50000
```

SALES detail data set entry.

```
54283545
4397D13P
1
4590
276
0
121575
122075
```

REPORTING

SECTION IV

QUERY provides several reporting techniques which enable you to examine the data base content without writing programs. The techniques are based on various forms of the LIST and REPORT commands which are described in this section.

The LIST command provides automatic formatting including column alignment and headings. The functions of selectively locating and reporting on entries are combined in a single command.

REPORT ALL, one form of the REPORT command, is similar to the LIST command since you do not specify a format. However, REPORT ALL prints all data item values in each entry located with the last FIND command. QUERY prints data item name and value pairs, one item per line, for each data item in each entry located.

If you want to design your own report format, you can use another form of the REPORT command which allows you to produce reports which include:

- From 1 to 9 lines of heading information such as a title, column headings, page numbers, date and time of day at the top of each report page. (You may also have blank lines interspersed in the heading but the heading cannot exceed one page.)

- Sorted entries with group (or subset) and total information printed with entries belonging to the same group. (You can accumulate totals, compute averages, and count entries automatically or use the ten QUERY registers to do other computations on numeric data item values and report the results.)

- Edited data item values with inserted dollar signs, minus signs, decimal points, and other ASCII characters. (You may also suppress leading zeros.)

You can include statements in your report which change the output device to the QSLIST device, define the number of lines per page, request a pause after each page is printed in session mode, and suppress the margins which normally appear at the top and bottom of a page (for example, information on a line printer can be printed across the perforation).

REPORT commands may be stored as command procedures.

LIST

Prints complete or partial entries from a single data set with automatic formatting and headings.

The form of the LIST command is

$$\left\{\begin{matrix}\text{LIST}\\ \text{L}\end{matrix}\right\} \left\{\begin{matrix}\textit{data set name}\\ \textit{data item list}\end{matrix}\right\} \left[\text{FOR } \textit{relation} \left[\left\{\begin{matrix}\text{AND}\\ \text{OR}\end{matrix}\right\} \textit{relation} \ldots\right]\right] \left[\text{END}\right]$$

For example,

LIST LABOR FOR BADGE# IE "09.18" AND F-NAME IE JOE

data set name — *relation* — *relation*

or

LIST BADGE#,F-NAME,L-NAME

data item list

or

LI LABOR

data set name

where

data set name	is the name of a data set in the current data base.
data item list	is a list of data item names separated by commas. All the data items must be from the same data set. You can list them in any order. A data item name may not be qualified with a data set name.
relation	takes the form: *data set name, data item name relop "value", "value"* . . .
data item name	is the name of a data item contained in the data set. If you use *data set name* instead of *data item list* preceding the FOR, you should not qualify any data item names with data set names. If you precede the FOR with a *data item list*, you can qualify the data item in the first relation with a data set name.

relop is a relational operator as shown in table 4-1.

value is a data item value. It must be the same type and within the same value range as the data item named in the *relation*. *Value* need not be enclosed in quote marks (") unless the value contains special characters. *Values* which are not contained in quotes are upshifted, for example, California is converted to CALIFORNIA before it is compared to data item values in the data base.

Table 4-1. LIST Command Relational Operators

OPERATOR	MEANING
= IS IE EQ	is equal to
# <> ISNOT INE NE	is not equal to
 ILT LT	is less than
>= INLT GE	is not less than (is greater than or equal to)
> IGT GT	is greater than
<= INGT LE	is not greater than (is less than or equal to)
IB *value*$_1$, *value*$_2$	is between (and including) *value*$_1$ and *value*$_2$

Multiple *values* may be used with these operators.

The operators <>, <=, >= cannot have any intervening spaces (embedded blanks).

LIST

The maximum number of logical connectors (AND, OR) which can be used in the LIST command is 10.

The LIST command prints all or a subset of the data item values from a single data set. It is one of the simplest ways to report on your data since you do not need to design a report or specify the format and headings.

LISTING FORMAT

The data is printed in columns. The width of a column (or field) is determined by the data item type. Table 4-2 summarizes field widths. QUERY provides two spaces between fields.

Data item names are printed as column headings at the top of each page. If the complete data item name is longer than the field width, it is truncated. Headings of character type data items are left-justified and numeric type items are right-justified.

If all of the data you request does not fit on one line (in one record), data items at the end of the *data item list* or the data entry are ignored. The line length varies with the device you are using. It is usually 72 to 80 characters for a terminal and a maximum of 136 characters for a line printer.

Table 4-2. Field Widths

ITEM TYPE	FIELD SIZE (in characters)
I1	6
I2	11
J1	5
J2	10
K1	5
R2	12
R3	17
Z*n*	n+1 (maximum=20)
P*n*	n (maximum=20)
U*n*	n (maximum=line length)*
X*n*	n (maximum=line length)*

*Absolute maximum=136, the line length for a line printer.

LISTING A SUBSET OF THE DATA

You may list a subset of the data set values in two ways:

- Use the *data item list* form of the command to specify particular items you want to list. For example,

  LIST F-NAME, L-NAME, YROFSERV

  lists the values of F-NAME, L-NAME, and YROFSERV for each entry in the set. This form is useful if the complete data entry does not fit on one line. You can change the order of the data items and print the last ones in the data entry first or print only the last items.

- Use the FOR parameter to set criteria for selection of entries from the set. For example,

  LIST LABOR FOR YROFSERV GE 5

  lists the value of all data items in each entry of the LABOR data set containing YROFSERV values greater than or equal to 5.

It is also possible to combine these techniques. For example,

LIST F-NAME, L-NAME FOR YROFSERV GE 5

lists the full names of each person entered in the set whose years of service (YROFSERV) total is greater than or equal to 5. Note that the data item used as selection criteria need not be in the data item list.

LIST

DETERMINING THE DATA SET TO BE LISTED

If you use the LIST command form specifying a data set name, there is no ambiguity as to which information will be listed. If you use a *data item list*, you should consider the following:

- If the command has a FOR clause, QUERY uses the data item in the first *relation* to determine the data set to be listed.
- If there is no FOR clause, it uses the last item in the *data item list.*

In either case, if the data item appears in only one data set, that set (or a subset of it) is listed. If it appears in more than one set and is qualified, the named data set is listed. Otherwise, QUERY uses the data set list and follows the rule described with the DATA-SET= command. (See Section II.)

THE RELATION OF LIST AND FIND

The entries selected by the LIST command are not available for any other purpose except the output of this command. The entries selected by the most recent FIND command are unaffected by LIST and are still available for use with the UPDATE and REPORT commands.

EXAMPLES

In the example below QUERY could not locate the required entry until the value was entered with the correct spacing.

```
>L CUSTOMER FOR STREET-ADDRESS="  868     DOYLE ROAD"

>L CUSTOMER FOR STREET-ADDRESS="868 DOYLE ROAD"

   ACCOUNT   LAST-NAME          FIRST-NAME   IN   STREET-ADDRESS

 10034765   SLATER             GINGER       K    868 DOYLE ROAD
```

In the next two examples, the data item values for INVENTORY and SALES entries with STOCK# equal to 6650D22S are listed.

```
>L INVENTORY FOR STOCK# =6650D22S

STOCK#      ONHANDQTY  SUPPLIER          UNIT-COS  LASTSH  BIN

6650D22S         5306  ACME WIDGET           1427  120375    3
6650D22S          600  HEWLETT-PACKARD      12500  111575    3
6650D22S            3  H & S SURPLUS            0  121575    0
6650D22S          999  H & S SURPLUS         1500  120575    0
6650D22S           13  H & S SURPLUS         1445  121475    3
6650D22S           11  H & S SURPLUS         1395  121575    3
6650D22S            3  H & S SURPLUS            0  121575    0
6650D22S           11  H & S SURPLUS         1395  121575    3
```

```
>L SALES FOR STOCK#=6650D22S

  ACCOUNT   STOCK#     QUANTI       PRICE          TAX          TOTAL  PURCH-

 24536173  6650D22S         3         598           20              0  120875
 24566356  6650D22S         1       12500          750              0  121575
```

```
>LIST CUSTOMER FOR ACCOUNT GT 55555555
```

List CUSTOMER entries with ACCOUNT greater than 55555555.

```
  ACCOUNT   LAST-NAME          FIRST-NAME  IN  STREET-ADDRESS

 76623455  MCFALL            JEFFREY     X   6650 MONTEREY ROAD
 74001813  FIELD             HUBERT      J   4556 GEARY
 87654321  JONES             JOHN        P   1 PINE AVE
 80808080  CELERY            ALLISON     B.  18 ASCOT AVE.
 77765555  PALMER            ERNEST      M   3728 CHECKERS COURT
 99998877  MEADOWS           JASPAR      A   5606 SUNNYHILLS DRIVE
```

LIST

```
>LIST LAST-NAME,STREET-ADDRESS,CITY,STATE FOR ACCOUNT GT 55555555
ACCOUNT          IS A MEMBER OF THESE SETS:
CUSTOMER,SALES
WHICH SET DO YOU WISH TO USE?
>>CUSTOMER
```

All values do not fit on the line so you can also list by data item name to get the items at the end of the entry.

```
LAST-NAME          STREET-ADDRESS                CITY          ST

MCFALL             6650 MONTEREY ROAD            CARMEL        CA
FIELD              4556 GEARY                    CUPERTINO     CA
JONES              1 PINE AVE                    CAMPBELL      CA
CELERY             18 ASCOT AVE.                 CARMEL        CA
PALMER             3728 CHECKERS COURT           ALVISO        CA


>LIST STOCK#
STOCK#           IS A MEMBER OF THESE SETS:
PRODUCT,SALES,INVENTORY
WHICH SET DO YOU WISH TO USE?
>>INVENTORY
```

INVENTORY is entered in the data set list.

```
STOCK#

6650D22S
2457A11C
3586T14Y

< CONTROL Y >
```

Control Y terminates the listing.

```
7391Z22F
```

```
>LIST STOCK# FOR QUANTITY IGT 2 AND ACCOUNT ILT 88888888

STOCK#

6650D22S
3586T14Y
5405T14F
78
< CONTROL Y >

>LIST STOCK#
STOCK#            IS A MEMBER OF THESE SETS:
SALES, INVENTORY
WHICH SET DO YOU WISH TO USE?
>>INVENTORY

STOCK#

6650D22S
2457A11C
3586T14Y
7391Z22F
5405T14F
6650D22S
6650D22S
7391Z22F
5405T14F
4397D13P
3739A14F
6650D22S
6650D22S
6650D22S
```

Since QUANTITY is in the SALES data set and is part of the first relation, SALES data set STOCK# values are printed.

SALES is automatically added to the data set list.

SALES and INVENTORY are both in the data set list so QUERY must prompt for which set you want.

REPORT ALL

Prints data item values of entries located by the last FIND command without formatting.

The form of the REPORT command when used for this purpose is

{REPORT / R} *[output control statements]* ALL *[,character]*

For example,

REP ALL,X

character

or

REPORT ALL

where	*output control statements*	alter the standard output parameters. (See output control statements in this section.)
	character	is any printable ASCII character. If included in the command, the data item names are omitted from the report. Otherwise, the data item name precedes each data item value.

Each form of the REPORT command is an extension of the FIND command in that it prints a report of the data entries located by the last FIND command.

REPORT output can be directed to any desired output device through MPE :FILE commands and the QUERY OUTPUT command. Consult the description of OUTPUT for more information.

When QUERY prints an unedited negative number, a special character in the rightmost digit substitutes for the minus sign. This special character varies according to the rightmost digit of the value. Table 4-3 shows which special characters are used to indicate negative numbers. For example, the number –104 is represented as 10M according to the table. If you are not using REPORT ALL, you may edit using REPORT command edit statements to print the data item value with a negative sign and the last digit from 0 to 9.

Table 4-3. Negative Number Representation

UNITS DIGIT	NEGATIVE REPRESENTATION
0	} (may vary with the terminal used)
1	J
2	K
3	L
4	M
5	N
6	O
7	P
8	Q
9	R

EXAMPLES

```
>REPORT ALL
RECORD HAS NOT YET BEEN FOUND
>FIND ALL LAST-NAME
USING SERIAL READ
13  ENTRIES QUALIFIED
>REPORT ALL
```

If you use REPORT ALL and you have not located any entries previously, QUERY prints an error message.

Once records have been found, REPORT ALL prints the value for each item to which you have access.

```
ACCOUNT          =54283540
LAST-NAME        =CORCORAN
FIRST-NAME       =CLIFFORD
INITIAL          =C
STREET-ADDRESS   =6105 VALLEY GREEN DR.
CITY             =CARMEL
STATE            =CA
ZIP              =93921
CREDIT-RATING    = 7.10000


ACCOUNT          =54
< CONTROL Y >
```

REPORT ALL

```
>REPORT ALL,X
```

REPORT ALL with a character prints the data item values without the data item names.

```
54283540
CORCORAN
CLIFFORD
C
6105 VALLEY GREEN DR.
CARMEL
CA
93921
 7.10000


54283545
MAYFIELD
WILLIAM

37 41ST AVE.
PETALUMA
GA
10101
 8.50000


10293847

< CONTROL Y >
```

You can terminate the report at any time by entering Control Y.

REPORT

Lists data item values of entries located by the FIND command in the format you specify.

The form of the REPORT command is

{REPORT / R} *report statements* END

For example,

```
REPORT
  H1, "NAME LIST",20,SPACE A2      <-- header statement
  D,LAST-NAME,20                   <-- detail statements
  D,FIRST-NAME,30                  <--
  END
```

or

```
R H1, "NAME LIST", 20, SPACE A2;D,LAST-NAME,20;D,FIRST-NAME,30;END
```

header statement — *detail statements*

where *report statements* consist of a sequence of header, detail, sort, group, total, register, edit, and output control statements, as outlined in table 4-4. These statements are explained in detail later in this section. Statements can be entered on separate lines or on one line separated by semicolons.

Table 4-4. REPORT Statements

STATEMENT TYPE	FUNCTION
Header	Prints title, column headings, page numbers, time of day, and the date at the top of each report page.
Detail	Prints data item values in the column position specified.
Sort	Sorts data entries based upon the value of a specified data item.
Group	Prints a data item value or character string whenever the value of an appropriate "sort item" changes.
Total	Prints column count, average, or totals for logical groups or entire report.
Edit	Describes edit masks used to punctuate Group, Detail, or Total fields.
Register	Specifies an operation to be executed in Register *n*.
Lines =	Specifies the number of lines per page.
Nopage	Suppresses page advancing.
Out=LP	Switch output device.
Pause	Causes output to pause after each page.

REPORT

DESIGNING A REPORT

Report formats vary according to their use. However, many reports assume the general format depicted in figure 4-1. The TITLE and HEADERS describe the report and are printed at the top of each page along with the page number. HEADERS are usually used to describe the report columns.

The report body consists of DETAIL lines, GROUP TITLES, and TOTALS along with other descriptive labels. Normally, each detail line displays information from a single data entry, although information can appear on more than one line per entry. A DETAIL field can be edited to include commas, decimal points, dollar signs, and other punctuation characters.

DETAIL lines can be sorted and grouped according to the values of data items in the entry. For example, a sales report may list sales results by country, region, sales office, and finally by individual salesman within each office. A GROUP TITLE can be printed whenever a "sort field" changes value. For example, when the country changes, the name of the country could be displayed as a GROUP TITLE. The title can be a series of characters or a data item value.

SUBTOTALS can be printed for logical groups (for example, for each sales office) and GRANDTOTALS for the entire report. These totals add, average, or count the DETAIL fields in each column of the report. Like DETAIL and GROUP fields, TOTAL fields can be edited with punctuation characters.

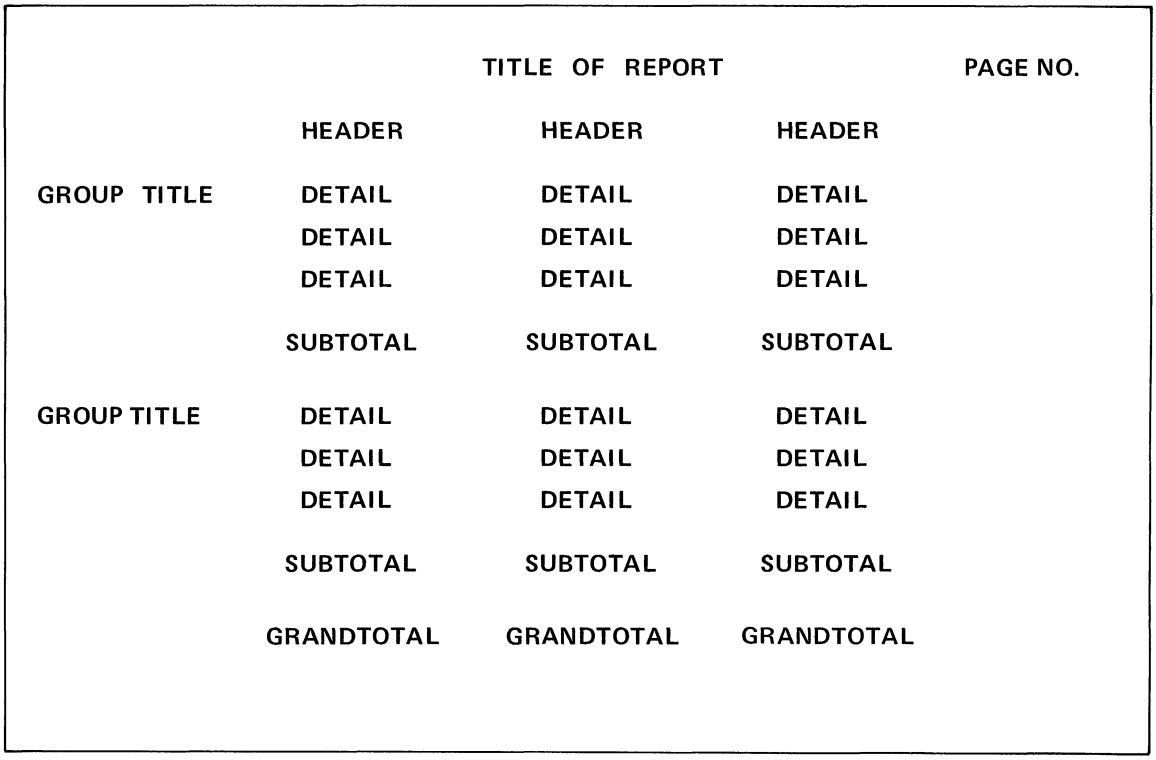


Figure 4-1. General Report Format

USING REPORT STATEMENTS

In the report statement descriptions which follow, a report is created by adding one statement type at a time and showing how the added statements change the report. Figure 4-2 contains the final version of the report.

```
AS OF: 01/14/76                                              PAGE 1
                        BOBO'S MERCANTILE
                        ON HAND INVENTORY

BIN#  SUPPLIER              STOCK        SHIP DATE      INVENTORY
                                                          AMOUNT

  0   H & S SURPLUS
                          7391Z22F       8/13/74        $5,012.50
                          5405T14F       9/11/74       $12,129.60
                          6650D22S      12/05/75       $14,985.00

      BIN TOTAL                                         $32,127.10 *

  1   ACME WIDGET
                          2457A11C      12/01/75  $553,477,666.95
      BAY PAPER CO.
                          7391Z22F      12/01/75        $4,704.00
      CARDINAL MILLS
                          5405T14F      11/28/75        $1,396.00
      JAKE'S JUNK
                          3739A14F      12/15/75        $1,169.32

      BIN TOTAL                                   $553,484,956.27 *

  2   ACME WIDGET
                          4397D13P       3/02/58       $55,080.00
      CARDINAL MILLS
                          3586T14Y      11/20/75          $358.56

      BIN TOTAL                                         $55,438.56 *

  3   ACME WIDGET
                          6650D22S      12/03/75       $75,716.62
      H & S SURPLUS
                          6650D22S      12/14/75          $187.85
                          6650D22S      12/15/75          $153.45
      HEWLETT-PACKARD
                          6650D22S      11/15/75       $75,000.00

      BIN TOTAL                                        $151,057.92 *


      TOTAL INVENTORY                             $553,723,579.85 **
```

Figure 4-2. Sample Report

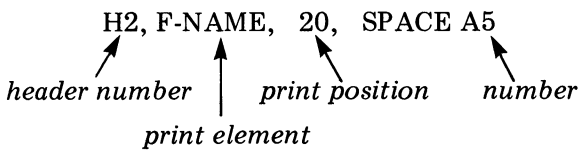
HEADER STATEMENTS

Header statements are used to print report titles and column headings at the top of each report page.

The form is

H *header number*, *print element*, *print position* [,SPACE A [*number*]]
[,SPACE B *[number]*]

For example,



where

header number — is an integer from 1 to 9. Up to nine lines of header information can be printed in addition to blank lines created by spacing before and after non-blank lines. Header information with the same header number is printed on the same line. The lowest-numbered header statement is printed first, the next-highest numbered statement is printed next. Header statements do not have to be consecutively numbered.

print element — is either PAGENO, DATE, TIME, a series of characters enclosed in quotes, or a data item name.

PAGENO numbers consecutively each page of the report.

DATE prints the date in the form: MM/DD/YY.

TIME prints the time in the form: HH:MM:SS.

Characters are stripped of the surrounding quotes and printed. The value of the specified data item is printed.

print position
SPACE A *number*
SPACE B *number*

See table 4-7 for descriptions of these parameters.

A header may contain up to 9 lines of information and any number of blank lines as long as it does not exceed the page size as defined by the output control statement, LINES=. (See the description at the end of this section.)

EXAMPLE

```
>F ALL INVENTORY.STOCK#
USING SERIAL READ
13  ENTRIES QUALIFIED
>REPORT
>>H1,"AS OF:",6
>>H1,DATE,15
>>H1,PAGENO,71
>>H1,"PAGE",69
>>H2,"BOBO'S MERCANTILE",45
>>H3,"ON HAND INVENTORY",45,SPACE A2
>>H7,"BIN#",4
>>H7,"SUPPLIER",14
>>H7,"STOCK",33
>>H7,"SHIP DATE",49
>>H7,"INVENTORY",68
>>H8,"AMOUNT",68
>>END
```

Locate entries.

Enter report statements describing the report headings. Character literals are printed as they appear in the statement.

DATE and PAGENO are generated and printed by QUERY

```
        Col.   Col.          Col.          Col.   Col.          Col.  Col.
         6      15            33            45     49            68    71
         ↓      ↓             ↓             ↓      |             ↓     ↓
H1 → AS OF: Ø1/14/76                                          PAGE  1
H2 →                        BOBO'S MERCANTILE
H3 →                        ON HAND INVENTORY
                                                  ↓
H7 → BIN#   SUPPLIER        STOCK        SHIP DATE        INVENTORY
H8 →                                                        AMOUNT
```

REPORT

DETAIL STATEMENTS

Detail statements usually specify a data item name whose value changes with each data entry reported, although a fixed series of characters can be specified as well. The statement specifies the print position, top-of-form, line spacing, and applicable edit masks.

The form is

D*[detail number]*,*print element*,*print position* [,SPACE A*[number]*]

[,SPACE B*[number]*] [,SKIP $\left\{\begin{matrix}A\\B\end{matrix}\right\}$] [,E $\left\{\begin{matrix}number\\Z\end{matrix}\right\}$]

For example,

D2, BADGE#, 35, SKIPB, E8

detail number — *print element* — *print position* — *label*

or

D, R3, 15, SPACE A2

print element — *print position* — *number*

where *detail number* is an integer from 1 to 9. If the number is omitted, the *print element* is printed on a group line when any control break occurs. (See the discussion of both group and sort statements.) The lowest numbered statement is printed first and others follow in numeric sequence. Detail statements with the same number are printed on the same line. Information from a single data entry can therefore be printed on up to ten separate lines.

print element is either a data item name, a register statement (R*n*) number, or a series of characters enclosed in quotes. A series of characters is printed without the surrounding quotes. It is printed once for each entry reported. If a data item is specified its value is printed for each entry reported. If a register statement number is specified, the data in the register is printed. You must enter both the letter R and number *n*. See the description of the R*n* statement which follows.

print position
SPACE A *number*
SPACE B *number*
SKIP {A | B}
E {*number* | Z}

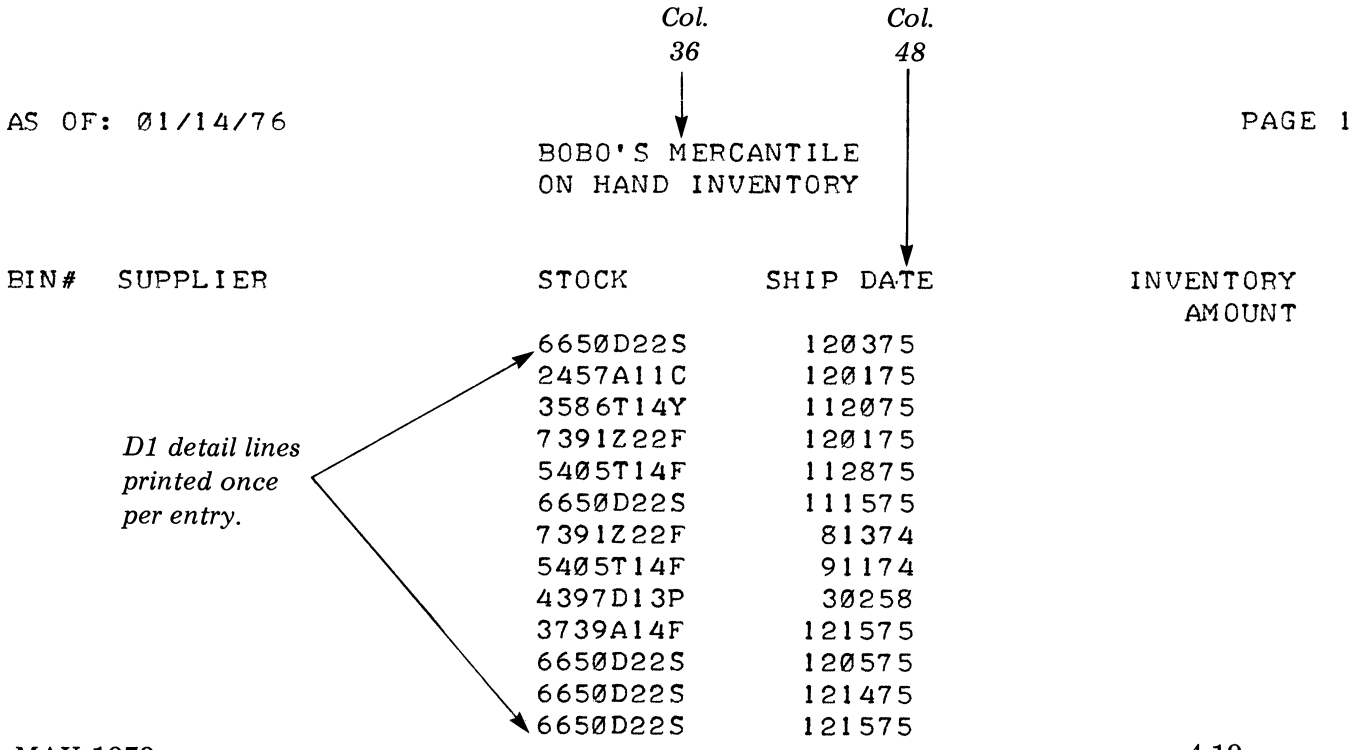
See table 4-7 for descriptions of these parameters and the edit statement which follows.

EXAMPLE

```
>REPORT
>>H1,"AS OF:",6
>>H1,DATE,15
>>H1,PAGENO,71
>>H1,"PAGE",69
>>H2,"BOBO'S MERCANTILE",45
>>H3,"ON HAND INVENTORY",45,SPACE A2
>>H7,"BIN#",4
>>H7,"SUPPLIER",14
>>H7,"STOCK",33
>>H7,"SHIP DATE",49
>>H7,"INVENTORY",68
>>H8,"AMOUNT",68
>>D1,STOCK#,36
>>D1,LASTSHIPDATE,48
>>END
```

If the same report shown in the header statement example contains detail statements, the result is as shown below.

This detail statement prints the value of STOCK# ending in column 36. The next statement prints LASTSHIPDATE value on the same detail line ending in column 48.



REPORT

EDIT STATEMENTS

The edit statement is used for punctuating data item values printed in a report. The statement performs such functions as suppressing leading zeros in numeric values, inserting characters such as dollar signs, dashes, commas, and decimal points. It also masks characters to eliminate them from the printed output. Edit statements may appear anywhere in the report.

The form is

E *number, "edit mask"*

For example,

E3, "$$$999CR"

number *edit mask*

where *number* is an integer from 0 to 9 identifying the edit statement. A report may have at most ten edit statements, each with a unique number.

edit mask consists of from 1 to 20 characters (if the mask is to edit numeric data item values) or from 1 to the maximum record length of the output device (if the mask is to edit alphanumeric data item values). In either case, the characters must be bracketed by quotes. The length of the edit mask determines the length of the output field that is printed.

ALPHANUMERIC EDIT MASKS. Alphanumeric edit masks consist of X's (used as place holders) and any other ASCII printing characters (used as insertion characters). QUERY examines the data item value specified in the detail, group, or total statement and the edit mask specified in the referenced edit statement, starting with the leftmost character of each.

If the character in the edit mask is X a character from the data item value is printed in the corresponding position of the output field. If the character in the edit mask is any character other than an X, the edit mask character is printed in the corresponding position of the output field. For example, if the value ABCD is edited with the mask "X-X-X-X", the result is printed as A-B-C-D.

If there are fewer X's in the edit mask than there are characters in the data item value, the rightmost characters of the data item value that do not correspond to an X in the mask are omitted. For example, if the value ABCD is edited with the mask "XX", the result is printed as AB.

If there are more X's in the edit mask than there are characters in the data item value, QUERY prints asterisks in place of the unused X's in the edit mask. All insertion characters in the mask are printed in the output field. For example, if the value ABCD is edited with the mask "XXXX-X", the result is printed as ABCD-*.

Here are two more examples of alphanumeric edit masks:

Data Item Value	Edit Mask	Printed Result
A34B	"X//X-X-X-X"	A//3-4-B-*
ABCD	"- - - X"	- - - A

NUMERIC EDIT MASKS. A numeric edit mask consists of the placement holders (9, Z, *, $), the sign characters (CR,-), and any other ASCII printing characters used as insertion characters. Each of the place holders and sign characters serves a special purpose in editing data item values. The characters and their meanings are specified in table 4-5.

The numeric edit mask edits decimal integer values consisting of up to 20 characters (not counting the sign characters) in the combinations outlined in table 4-6.

If the number of significant digits of the data item value is greater than the number of place holders (9, Z, *, $) in the edit mask, the output field is filled with asterisks. For example, if the value 12345 is edited with the mask "999CR" the result is *****.

Only one decimal point may appear in any edit mask.

If a minus sign appears in the mask in any position other than the rightmost character of the mask, the minus is treated as an insertion character. For example, if the value 12345 is edited with the mask 999-99, the result is 123-45.

Figure 4-3 shows the results of printing numeric data item values using numeric edit masks.

REAL NUMBERS. Real values (type R2 and R4) cannot be edited. They are printed in either fixed (xx.xxx) or floating point (xx.xxxE± xx) form, depending upon the magnitude of the value. R2 values less than .1 or greater than 10^6 and R4 values less than .1 or greater than 10^{16} are printed in floating-point notation. All other R2 and R4 values are printed in fixed-point notation.

Fixed R2 values occupy up to eight characters and floating-point R2 values up to 12 characters. R4 values occupy up to 18 and 22 characters for fixed and floating-point respectively. For example:

Real Value	Type	Report Output
0	R2	.000000E+00
8399607	R2	.839961E+07
-.863617E-77	R2	-.863617E-77
-3.34567	R2	-3.34567
-.000000034567	R2	-.345670E-07
12345678987654321234567	R4	.1234567898765432E+23
.000333444555	R4	.3334445550000000E-03

Table 4-5. Numeric Edit Mask Characters

CHARACTER	EXPLANATION
9	Each 9 in the edit mask is replaced with a decimal digit from the data item value in the corresponding position of the output field.
Z	Z is a zero suppression place holder. A Z in the edit mask is replaced with a decimal digit from the data item value in the corresponding position of the output field. If the data item value digit under consideration is a zero appearing to the left of the leftmost significant digit, QUERY inserts a blank in the output field, and all other zeros to the left of the significant digit are replaced by a blank in the output field.
*	* is an asterisk place holder. An * in the edit mask acts just like a Z with the exception that leading zeros in the data item value are replaced with asterisks in the output field.
$	$ is the dollar sign place holder. A $ in the edit mask acts just like a Z, except that the first zero in the data item value to the left of the leftmost significant digit is replaced with the dollar sign in the output field. All zeros to the left of the first leading zero are replaced with blanks in the output field.
CR	CR is a sign character, and always appears in the two rightmost positions of the edit mask. If the data item value is negative, QUERY prints the two characters (CR) in the first two rightmost positions of the output field. If the data item value is positive, QUERY prints two blank characters in place of the CR. No characters from the data item value are ever placed in the first two rightmost positions of the output field.
-	- is a sign character and acts the same as the CR characters. If the data item value is negative, QUERY prints the minus sign (-) in the rightmost position of the output field. If the data item value is positive, QUERY prints a blank in place of the minus. No character from the data item value is ever placed in the rightmost position of the output field.
Insertion characters	Insertion characters consist of any ASCII printing characters not previously mentioned. Insertion characters are printed in the output field in the same position they appear in the edit mask. Any insertion character appearing in the edit mask to the left of the leftmost significant digit of the data item value is replaced with blanks or an asterisk, depending upon which zero suppression character is specified in the edit mask. Only one decimal point can appear in an edit mask.

Table 4-6. Numeric Edit Mask Combinations

COMBINATIONS	EXAMPLE
9's only	"99" "999999" "9999"
Z's only	"ZZ" "ZZZZZZZZZZ"
Asterisks only	"****" "**********"
Dollar signs only	"$$$$" "$$$$$$$$$$$$$"
Sign characters preceded by 9's	"9999CR" "999999-"
Sign characters preceded by 9's which are preceded by Z's, asterisks, or dollar signs	"$$$999CR" "ZZ99999-" "******9999999CR"
9's preceded by Z's, asterisks, or dollar signs	"$$$999999" "ZZZZ9999" "***99999999"
Any of the above combinations including insertion characters such as commas, one decimal point, slashes, etc. located anywhere in the edit mask, except to the right of sign characters.	"$$999,999.99-" "999-9999" "99/99/99" "$$9999.99CR"

DATA ITEM VALUE	EDIT MASK	PRINTED RESULT
0059	"$$$,999"	$059
001024	"ZZZ,ZZZ"	1,024
-0010555	"$$,$$$.99CR"	$105.55CR
00010555	"$$,$$$.99CR"	$105.55
-0010555	"$$,$$$.99-"	$105.55-
00010555	"$$,$$$,99-"	$105.55
15039250	"$,$$$,$$$.99CR"	$150,392.50
00049	"*****"	***49
044240474	"999-99-9999"	044-24-0474
-2145	"$,$$$.99"	$21.45

Figure 4-3. Sample Output Using Numeric Edit Masks

REPORT

EXAMPLE

```
>REPORT
>>H1,"AS OF:",6
>>H1,DATE,15
>>H1,PAGENO,71
>>H1,"PAGE",69
>>H2,"BOBO'S MERCANTILE",45
>>H3,"ON HAND INVENTORY",45,SPACE A2
>>H7,"BIN#",4
>>H7,"SUPPLIER",14
>>H7,"STOCK",33
>>H7,"SHIP DATE",49
>>H7,"INVENTORY",68
>>H8,"AMOUNT",68
>>D1,STOCK#,36
>>D1,LASTSHIPDATE,48,E2
>>E2,"XX/XX/XX"
>>END
```

Edit LASTSHIPDATE with mask E2.
Define mask E2.

Now the report looks like this:

```
 OF: 01/14/76                                                    PAGE 1
                          BOBO'S MERCANTILE
                          ON HAND INVENTORY

N#   SUPPLIER              STOCK        SHIP DATE          INVENTORY
                                                              AMOUNT
                          6650D22S     12/03/75
                          2457A11C     12/01/75
                          3586T14Y     11/20/75
                          7391Z22F     12/01/75
                          5405T14F     11/28/75
                          6650D22S     11/15/75
                          7391Z22F      8/13/74
                          5405T14F      9/11/74
                          4397D13P      3/02/58
                          3739A14F     12/15/75
                          6650D22S     12/05/75
                          6650D22S     12/14/75
                          6650D22S     12/15/75
```

SORT STATEMENTS

The sort statement serves two purposes in the REPORT command; it

- specifies data items whose values are used to sort data entries when they are printed in the report,
- defines control break levels for use by group and total statements in the report.

The form is

S*[level]*,*data item name* [, {ASC / DES}]

For example,

```
S3,  BADGE#  ,ASC
 ↑      ↖
level   data item name
```

or

```
S,  L-NAME
     ↑
data item name
```

where *level* is an integer from 1 to 5. A sort statement with *level* greater than 1 must be accompanied by sort statements containing all lower level numbers. That is, if S3 appears in the report, S2 and S1 must also appear in it. You need not specify a level; in this case, the sort statement sorts but does not define a control break.

data item name is the name of a data item contained in data entries selected by the last FIND command. If you last used FIND CHAIN, you may not use a *master* data item name in the sort statement.

DES indicates that the data item values are to be ordered in descending order.

ASC indicates that the data item values are to be ordered in ascending order. If you do not specify ASC or DES, the default order is ASC.

The sort statement:

```
S1,L-NAME
```

sorts the entries selected into the order specified by the value of L-NAME.

Data Entries After FIND

L-NAME	F-NAME	AGE
WHITE	ROB	26
BROWN	JACK	32
GREEN	ROB	49
WHITE	LARRY	81
BROWN	CHRIS	17
GREEN	SAM	28
GREEN	BILL	45
BROWN	DAN	39
WHITE	WILL	22

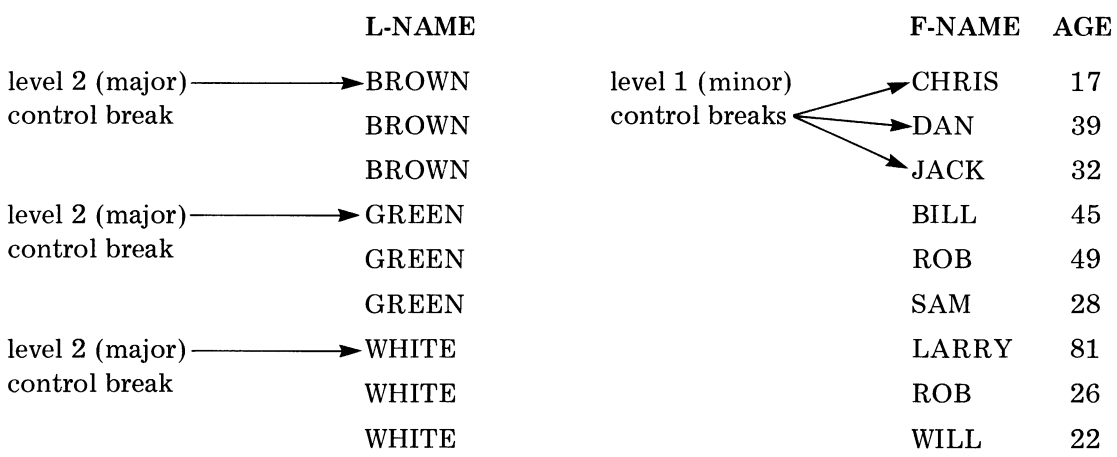
Data Entries After Sort Executed

L-NAME	F-NAME	AGE
BROWN	JACK	32
BROWN	CHRIS	17
BROWN	DAN	39
GREEN	ROB	49
GREEN	SAM	28
GREEN	BILL	45
WHITE	ROB	26
WHITE	LARRY	81
WHITE	WILL	22

You can sort on many different data items. The higher-numbered sort statement identifies the major (or first) sort field, while the lower-numbered sort statement identifies the minor sort field. The minor sort arranges entries in the order specified, keeping all major sort items with identical values together, in other words, it sorts within subsets of the entire set of entries. For example, if the statement illustrated above appeared in a report with another statement:

```
S1,F-NAME
S2,L-NAME
```

the result would be as follows:



CONTROL BREAKS. A control break occurs during the printing of a report whenever the current entry's value for a data item defined in a numbered sort statement is different from the last entry's value. When the first entry is printed, a control break occurs since the data item value changes from null (no value) to the first value. Totals are not printed when the first control break occurs.

In the examples above, a control break occurs when the value of L-NAME becomes BROWN and when it changes to GREEN, and again when it changes to WHITE. This is known as a level 2 control break because the data item named L-NAME appears in a sort statement labeled S2. The level 1 control break is associated with the data item named F-NAME and sort statement labeled S1.

A control break occurs for all lower levels whenever a higher level control break occurs. This means that whenever a control break occurs for level 2 (L-NAME), a control break occurs for level 1 (F-NAME) by definition.

A group or total statement prints only when a control break occurs that is at the same level as the group or total statement. This means that a total statement labeled T1 prints only when a level 1 control break occurs, or a group statement labeled G2 prints only when a level 2 control break occurs. Consult the descriptions of group and total statements below for explanation of their functions.

Sort statements with no *level* (i.e., no number) are used to sort entries but do not define control breaks for use by group or total statements.

MAJOR TO MINOR SORT FIELDS. Numbered and unnumbered sort statements can appear in the same REPORT command. The order in which unnumbered sort statements appear in the *report body* is significant. The first unnumbered statement defines the most minor sort field, while the last unnumbered statement defines the most major sort field. QUERY defines sort fields in the following order from most major to most minor:

Most Major						Most Minor
S5	S4	S3	S2	S1	S(last in *report body*) . . . . .S(first in *report body*)	

For example, the sort statements below define the order as shown:

Statements	Order		
S2,OFFICE	MONTH	← major	(S3)
S,PARTNO	OFFICE		(S2)
S1,SLSMAN	SLSMAN		(S1)
S,QUANTITY	QUANTITY		(S last)
S3,MONTH	PARTNO	← minor	(S first)

MAXIMUM NUMBER OF SORT ITEMS. The number of data items you may use to sort is limited in two ways:

- the combined length of the data items appearing in all the sort statements must not exceed 2045 words,
- the maximum number of sort statements allowed in a single report is 66. In other words, you may have up to 66 sort statements provided that the combined length of the data items in all of the 66 statements is not greater than 2045.

EXAMPLES

```
>REPORT
>>H1,"AS OF:",6
>>H1,DATE,15
>>H1,PAGENO,71
>>H1,"PAGE",69
>>H2,"BOBO'S MERCANTILE",45
>>H3,"ON HAND INVENTORY",45,SPACE A2
>>H7,"BIN#",4
>>H7,"SUPPLIER",14
>>H7,"STOCK",33
>>H7,"SHIP DATE",49
>>H7,"INVENTORY",68
>>H8,"AMOUNT",68
>>D1,STOCK#,36
>>D1,LASTSHIPDATE,48,E2
>>E2,"XX/XX/XX"
>>S2,BINNUM
>>S1,SUPPLIER
>>S,LASTSHIPDATE
>>END
```

Define BINNUM as sort level 2, SUPPLIER as sort level 1, and LASTSHIPDATE as sort without control break.

The detail entries are now sorted by SUPPLIER and BINNUM. The values for these items will be printed in group statements which are described next.

```
AS OF: Ø1/14/76                                                    PAGE 1

                              BOBO'S MERCANTILE
                              ON HAND INVENTORY
BIN#   SUPPLIER               STOCK        SHIP DATE          INVENTORY
                                                                 AMOUNT
                              7391Z22F      8/13/74
                              5405T14F      9/11/74
                              6650D22S     12/05/75
                              2457A11C     12/01/75
                              7391Z22F     12/01/75
                              5405T14F     11/28/75
                              3739A14F     12/15/75
                              4397D13P      3/02/58
                              3586T14Y     11/20/75
                              6650D22S     12/03/75
                              6650D22S     12/14/75
                              6650D22S     12/15/75
                              6650D22S     11/15/75
```

GROUP STATEMENTS

A group statement prints the value of a data item, the value in a register (R*n*), or a series of characters whenever a control break occurs.

The form is

G *level, print element, print position* [,SPACE A *[number]*] [,SPACE B *[number]*]

[,SKIP {A / B}] [,E {*number* / Z}]

For example,

```
G3, WEEK, 35, SKIP B, E2
 level   print element   print position   number
```

or

```
G1, R3, 55 ,SPACE B2
 level   print element   print position   number
```

where *level* is an integer from 1 to 5 corresponding to the *level* of a sort statement.

print element is either a data item name, a register statement number (R*n*), or a series of characters enclosed in quotes. (The quotes are stripped when the characters are printed.)

print position
SPACE A *number*
SPACE B *number*
SKIP {A / B}
E {*number* / Z}

See table 4-7 for descriptions of these parameters. Also see the edit statement description in this section.

REPORT

Each control break occurs as a result of a sort statement labeled from 1 to 5. When the control break occurs, the group statement with the same number as the sort statement prints the information you specify. (See the explanation of control breaks given under sort statements for more information.)

Whenever a control break occurs, all group statements with a number equal to or less than the level of the sort statement causing the break print a value or series of characters. All group statements print on the same line.

Since a control break always occurs at the very beginning of the report, all group statements print their contents before any detail statements are executed.

If the REPORT command contains group statements but no sort statements, an error is printed when the command is completed.

EXAMPLES

```
>REPORT
>>H1,"AS OF:",6
>>H1,DATE,15
>>H1,PAGENO,71
>>H1,"PAGE",69
>>H2,"BOBO'S MERCANTILE",45
>>H3,"ON HAND INVENTORY",45,SPACE A2
>>H7,"BIN#",4
>>H7,"SUPPLIER",14
>>H7,"STOCK",33
>>H7,"SHIP DATE",49
>>H7,"INVENTORY",68
>>H8,"AMOUNT",68
>>D1,STOCK#,36
>>D1,LASTSHIPDATE,48,E2
>>E2,"XX/XX/XX"
>>S2,BINNUM
>>S1,SUPPLIER
>>S,LASTSHIPDATE
>>G2,BINNUM,3,SPACE B
>>G1,SUPPLIER,20
>>END
```

BINNUM will be printed when a control break occurs for sort level 2, and
SUPPLIER when a sort level 1 control break occurs.

REPORT

AS OF: Ø1/14/76 PAGE 1

BOBO'S MERCANTILE
ON HAND INVENTORY

BIN#	SUPPLIER	STOCK	SHIP DATE	INVENTORY AMOUNT
Ø	H & S SURPLUS			
		7391Z22F	8/13/74	
		54Ø5T14F	9/11/74	
		665ØD22S	12/05/75	
				← *level 1 and level 2 control break*
1	ACME WIDGET			
		2457A11C	12/Ø1/75	
	BAY PAPER CO.			
		7391Z22F	12/Ø1/75	
	CARDINAL MILLS			← *level 1 control break*
		54Ø5T14F	11/28/75	
	JAKE'S JUNK			
		3739A14F	12/15/75	
				← *level 1 and level 2 control break*
2	ACME WIDGET			
		4397D13P	3/Ø2/58	
	CARDINAL MILLS			
		3586T14Y	11/2Ø/75	
3	ACME WIDGET			
		665ØD22S	12/Ø3/75	
	H & S SURPLUS			
		665ØD22S	12/14/75	
		665ØD22S	12/15/75	
	HEWLETT-PACKARD			
		665ØD22S	11/15/75	

Notice that the two data items mentioned in the detail statements do not print on the same line as the group statements. This is because the detail statements are numbered. Unnumbered detail statements print on the same line as a group statement whenever a control break occurs.

REPORT

TOTAL STATEMENTS

The total statement prints a data item value, the value in a register (R*n*), a series of characters, the total, average, or count of a group of data items whenever a control break occurs.

The form is

T *level,print element,print position* [,SPACE A *[number]*]

$$[,\text{SPACE B } [\textit{number}]\ ]\ \left[,\text{SKIP}\ \begin{Bmatrix}\text{A}\\ \text{B}\end{Bmatrix}\right]\ \left[,\text{E}\begin{Bmatrix}\textit{number}\\ \text{Z}\end{Bmatrix}\right]\ \left[,\ \begin{Bmatrix}\text{ADD}\\ \text{AVERAGE}\\ \text{COUNT}\end{Bmatrix}\right]$$

or the special form

T *level*, R*n*

For example,

T4, WAGES, 60, SKIP A, E2, ADD

level *print element* *print position* *number*

or

T1, "TOTAL WAGES=", 40, SPACE B5

level *print element* *print position* *number*

or

T3, R2

level *n*

where *level* is the letter F or an integer from 1 to 5 corresponding to the *level* of a sort statement. The letter F indicates the information is to be printed only at the end of the report after the last detail line and it relates to the entire report, not just a subgroup.

print element is either a data item, a register statement number (R*n*), or a series of characters enclosed in quotes (a character literal). The characters are printed without the surrounding quotes.

print position SPACE A *number* SPACE B *number* SKIP A / B E { *number* / Z }	See table 4-7 for definitions of these parameters. Also see the description of the edit statement.
ADD	indicates that you want to print the control group total of the values for the data item specified as the *print element.*
AVERAGE	indicates you want to print the control group average value for the data item specified as the *print element.*
COUNT	indicates you want to print a count of the number of values for that control group.
R*n*	is the number of a register to be cleared.

If the total statement is labeled TF, the ADD, AVERAGE, and COUNT options apply to all occurrences of the data item in the report. The options ADD and AVERAGE are not allowed on ASCII (U or X type) data items. If ADD, AVERAGE, or COUNT is used, the *print element* must be a data item name.

If you use the special form of the command, specifying only a register number without a *print element*, the register is cleared (reset to zero) when a control break occurs.

A control break results from a sort statement labeled from 1 to 5. When a control break occurs, the total statement corresponding to the sort level causing the break prints the information you specify. (See the description of the sort statement for more information about control breaks.)

Total statements must be accompanied by sort statements or an error is printed when the REPORT command is completed.

To perform more than one operation on the same data item (total, average, count) you must specify a separate total statement for each operation. The desired information is printed on the same line ending in the column position specified if you use the same *level* for each statement.

REPORT

EXAMPLE

```
>REPORT
>>H1,"AS OF:",6
>>H1,DATE,15
>>H1,PAGENO,71
>>H1,"PAGE",69
>>H2,"BOBO'S MERCANTILE",45
>>H3,"ON HAND INVENTORY",45,SPACE A2
>>H7,"BIN#",4
>>H7,"SUPPLIER",14
>>H7,"STOCK",33
>>H7,"SHIP DATE",49
>>H7,"INVENTORY",68
>>H8,"AMOUNT",68
>>D1,STOCK#,36
>>D1,LASTSHIPDATE,48,E2
>>E2,"XX/XX/XX"
>>S2,BINNUM
>>S1,SUPPLIER
>>S,LASTSHIPDATE
>>G2,BINNUM,3,SPACE B
>>G1,SUPPLIER,20
>>T2," *",70
>>T2,"BIN TOTAL",14,SPACE A
>>TF,"TOTAL INVENTORY",20,SPACE B3
>>TF," **",71
>>END
```

This statement prints an asterisk in column 70 and the next one prints a character literal when a level 2 control break occurs.

When the final totals are printed, these statements print the specified characters.

The total statements which have been added merely print character literals. The totals are computed with register statements which are described next. More total statements are added in the example of register statements.

REPORT

AS OF: 01/14/76 PAGE 1

BOBO'S MERCANTILE
ON HAND INVENTORY

BIN#	SUPPLIER	STOCK	SHIP DATE	INVENTORY AMOUNT	
0	H & S SURPLUS				
		7391Z22F	8/13/74		
		5405T14F	9/11/74		
		6650D22S	12/05/75		
	BIN TOTAL		*level 2 control break*		*
1	ACME WIDGET				
		2457A11C	12/01/75		
	BAY PAPER CO.				
		7391Z22F	12/01/75		
	CARDINAL MILLS				
		5405T14F	11/28/75		
	JAKE'S JUNK				
		3739A14F	12/15/75		
	BIN TOTAL				*
2	ACME WIDGET				
		4397D13P	3/02/58		
	CARDINAL MILLS				
		3586T14Y	11/20/75		
	BIN TOTAL				*
3	ACME WIDGET				
		6650D22S	12/03/75		
	H & S SURPLUS				
		6650D22S	12/14/75		
		6650D22S	12/15/75		
	HEWLETT-PACKARD				
		6650D22S	11/15/75		
	BIN TOTAL				*
	TOTAL INVENTORY				**

REPORT

REGISTER STATEMENTS

A register statement specifies an operation to be executed in one of ten QUERY registers. The register statements are executed sequentially as they appear in the REPORT command, once for each data entry in the report (that is, once for each entry selected by the last FIND command.)

The form is

R *number*, { L [OAD] / A [DD] / S [UBTRACT] / M [ULTIPLY] / D [IVIDE] } , *data element*

For example,

```
R3, ADD,PRICE
 ↑          ↖
number      data element
```

or

```
R0, M, "25"
 ↑       ↖
number   data element
```

or

```
R5, DIV, R6
 ↑        ↖
number    data element
```

where

number	is an integer between 0 and 9 which identifies the register you want to use.
data element	is a numeric type data item name, a register number (R*n*), or a number enclosed in quotes (a numeric literal). The quotes are required. QUERY strips them and uses the number in the operation.
LOAD	indicates you want to replace the current contents (if any) of the register with the *data element*.
ADD	indicates you want to add the *data element* to the contents of the register.
SUBTRACT	indicates you want to subtract the *data element* from the contents of the register.

MULTIPLY	indicates you want to multiply the contents of the register by the *data element.*
DIVIDE	indicates you want to divide the contents of the register by the *data element.*

After each operation is executed, the result is placed in the register you specify at the beginning of the statement. For example, if R2 contains an integer 3, R4 contains an integer 2, and this statement is executed:

```
R2, MULTIPLY, R4
```

R2 will contain an integer 6.

USING QUERY REGISTERS. The register statements in a REPORT command describe a fixed sequence of operations to be performed each time a new data entry is processed for the report. If you are familiar with programming techniques, you may consider the R*n* statements as a program "loop" or routine which is executed once for each data entry included in the report.

R*n* statement execution affects only the ten QUERY registers. No output results from the statements although you may print the content of any register by using other REPORT command statements (with the exception of sort and header statements.)

All register operations except LOAD are cumulative. In other words, when each R*n* statement is executed, the current contents of the register are operated on by the data element and the result is stored in the register again.

INITIALIZING REGISTERS. Each register is initialized to zero when the REPORT command begins execution. You may reset a register to zero in two ways:

- load a zero into the register. For example,

  ```
  R3, L, "0"
  ```

- use a total statement to reset the register to zero when a control break occurs. For example, the statement:

  ```
  T2, R3
  ```

  sets Register 3 to zero when a control break occurs as a result of sort level 2.

An arithmetic operation may also reset the register to zero. For example,

```
R4, SUBTRACT, QUANTY        (where QUANTY is equal to the contents of R4)
```

If you divide the contents of a register by zero, the result is zero. Results of integer division are truncated.

REGISTER AND DATA TYPES. Only numeric type data can be used in register operations. The following IMAGE data item types are allowed:

- whole numbers or integers (I1, I2, J1, J2, K1, Z*n*, and P*n*)
- real numbers (R2)
- extended precision real numbers (R4).

You can mix data types in a register operation; for example, you can add an integer to a real number. QUERY determines the data type of the register content after such an operation by assigning an order of precedence (or rank) to the data types. The order is:

HIGHEST	R4 (Extended precision type data)
	R2 (Real type data)
LOWEST	I1, I2, J1, J2, K1, Z*n*, P*n* (Integers and packed decimal numbers)

The new register content always has a data type which is the higher of the two operand types: the old register content or the *data element.* For example, if Register 2 is loaded with an integer and then multiplied by a real type data item:

```
R2,L,"3"

R2,M,PERCENT        (where the value of PERCENT is 25.6)
```

the content of R2 will be type real after the statements have been executed.

The maximum size integer (including all IMAGE integer data types) which a register can contain is 19 digits.

Real and extended precision numbers have the same limits in registers as IMAGE R2 and R4 data items. R2 can have 6 to 7 significant digits and R4 can have 16 to 17 significant digits. If a register calculation results in overflow or underflow, a message is printed on the $STDLIST device.

When mixing data types in arithmetic register computations, you should think about the order of precedence and its effect on the calculations. For example, if you operate on a real register number with an integer having 12 significant digits, the result will have 6 to 7 significant digits.

NUMERIC LITERALS. To use a constant number in a register operation, you enter the number surrounded by quotes (for example, "325", ".0013", "-3E-6"). This type of number is called a numeric literal.

Integer numeric literals can have at most 19 digits. The length of real numeric literals is limited only by the line length (or input record length). Limits for real numeric literal values and significant digits are the R2 limits. Numeric literals may contain the following characters within the quotes:

- the digits 0 through 9 (integer and real)
- the plus (+) and minus (–) signs (integer and real)
- the letter E, upper or lower case (real only)
- the decimal point (real only).

It is not good practice to use blanks within numeric literals. They are rejected in register statements.

REPORT

EXAMPLE

```
>REPORT
>>H1,"AS OF:",6
>>H1,DATE,15
>>H1,PAGENO,71
>>H1,"PAGE",69
>>H2,"BOBO'S MERCANTILE",45
>>H3,"ON HAND INVENTORY",45,SPACE A2
>>H7,"BIN#",4
>>H7,"SUPPLIER",14
>>H7,"STOCK",33
>>H7,"SHIP DATE",49
>>H7,"INVENTORY",68
>>H8,"AMOUNT",68
>>D1,STOCK#,36
>>D1,LASTSHIPDATE,48,E2
>>D1,R7,68,E1
>>E2,"XX/XX/XX"
>>E1,"$$$,$$$,$$$,$$$.99"
>>S2,BINNUM
>>S1,SUPPLIER
>>S,LASTSHIPDATE
>>G2,BINNUM,3,SPACE B
>>G1,SUPPLIER,20
>>R7,LOAD,ONHANDQTY
>>R7,MULT,UNIT-COST
>>R8,ADD,R7
>>T2,R8,68,E1,SPACE B
>>T2," *",70
>>T2,R8
>>T2,"BIN TOTAL",14,SPACE A
>>TF,"TOTAL INVENTORY",20,SPACE B3
>>TF," **",71
>>TF,R9,68,E1,SKIP A
>>R9,ADD,R7
>>END
```

Add detail statement to print the content of Register 7 and edit with E1.

Load the value of ONHANDQTY into R7 and multiply by UNIT-COST.

Add the result to the contents of R8.

Add more total statements to print the content of Register 8 at each level 2 control break and clear Register 8.

Add a total statement to print the content of Register 9 on the final total line. R7 is added to R9 each time another entry's data is printed in the report.

Note: Each register statement is executed each time a new entry's data is printed in the report.

REPORT

AS OF: 01/14/76 PAGE 1

BOBO'S MERCANTILE
ON HAND INVENTORY

BIN#	SUPPLIER	STOCK	SHIP DATE	INVENTORY AMOUNT
		ONHANDQTY X UNIT-COST		
0	H & S SURPLUS			
		7391Z22F	8/13/74	$5,012.50
		5405T14F	9/11/74	$12,129.60
		6650D22S	12/05/75	$14,985.00
	BIN TOTAL	*Accumulated R7 values in R8. R8 set to 0*		$32,127.10 *
1	ACME WIDGET			
		2457A11C	12/01/75	$553,477,666.95
	BAY PAPER CO.			
		7391Z22F	12/01/75	$4,704.00
	CARDINAL MILLS			
		5405T14F	11/28/75	$1,396.00
	JAKE'S JUNK			
		3739A14F	12/15/75	$1,189.32
	BIN TOTAL			$553,484,956.27 *
2	ACME WIDGET			
		4397D13P	3/02/58	$55,080.00
	CARDINAL MILLS			
		3586T14Y	11/20/75	$358.56
	BIN TOTAL			$55,438.56 *
3	ACME WIDGET			
		6650D22S	12/03/75	$75,716.62
	H & S SURPLUS			
		6650D22S	12/14/75	$187.85
		6650D22S	12/15/75	$153.45
	HEWLETT-PACKARD			
		6650D22S	11/15/75	$75,000.00
	BIN TOTAL			$151,057.92 *
	TOTAL INVENTORY	*Accumulated R7 values*		$553,723,579.85 **

REPORT

OUTPUT CONTROL STATEMENTS

Output control statements may be included in a REPORT command to alter the standard parameters for report output.

There are four output control parameters. The form and purpose of each statement is:

LINES=*integer*	Specifies the number of lines per report page.
and	
NOPAGE	Suppresses page advancing at the beginning of each page (no margins are provided at the top and bottom of each page). All SKIP options are ignored, and all SPACE options are performed unconditionally.
and	
[OUT=] LP	Sends the report output to the QSLIST device. Applies only to the current report.
and	
PAUSE	Causes report output to pause after each page completes. Press *return* to continue. PAUSE is ignored in job mode or output is sent to QSLIST.

where *integer* specifies the number of lines per report page.

integer may be 0 which makes the page size infinite or between 10 and 32767.

The standard parameters for report output are:

- 60 lines per page
- page advancing at the beginning of each report page
- output printed on the $STDLIST device (the terminal in session mode and line printer in job mode).
- no pauses while report is being printed.

If you are using the REPORT ALL form of this command, these statements must precede the keyword ALL.

EXAMPLE

```
>F ALL LAST-NAME
USING SERIAL READ
13  ENTRIES QUALIFIED
>REPORT ALL
```

Locate all CUSTOMER entries.

Print them without output control statements.

```
ACCOUNT             =54283540
LAST-NAME           =CORCORAN
FIRST-NAME          =CLIFFORD
INITIAL
< CONTROL Y >
```

Terminate the list with Control Y.

Use the NOPAGE output control statement. QUERY does not skip lines for a top of page margin.

```
>REPORT NOPAGE;ALL
ACCOUNT             =54283540
LAST-NAME           =CORCORAN
FIRST-NAME          =CLIFFORD
INITIAL             =C
STREET-ADDRESS      =6105 VALLEY GREEN DR.
CITY
< CONTROL Y >

>REPORT OUT=LP;ALL
>REPORT LINES=8;ALL
NUMERIC VALUE ERROR
>REPORT
>>D1,LAST-NAME,20,SPACE A2
>>PAUSE
>>LINES=10
>>END
```

Specify output to the QSLIST device. Use the LINES= statement to set the number of lines per page (cannot be less than 10 unless 0).

Print a simple report with LINES=10 and a PAUSE statement.

```
    CORCORAN

    MAYFIELD

    WHITE

    DELLWIG


    FIELD
      •
      •
      •
```

After 10 lines, QUERY pauses and return must be pressed to continue the listing.

REPORT procedure

Executes a REPORT command stored on the current Proc-file as a procedure.

The form of the REPORT command when used for this purpose is

{REPORT / R} *[output control statements] procedure name [,character]*

For example,

```
REPORT  REP4,J
```

procedure name *character*

or

```
REP  TAX
```

procedure name

where

output control statements	see the output control statements in this section.
procedure name	is the name of a REPORT command stored as a procedure on the current Proc-file.
character	is any printable ASCII character. If *character* is included in the command, the REPORT procedure is listed before it is executed.

QUERY searches the current Proc-file and executes the REPORT command stored under the *procedure name.* The data entries used are those located by the previous FIND command. If the procedure does not exist on the current Proc-file, or if the Proc-file has not been declared, QUERY prints an error message.

Before the procedure is executed, QUERY checks it for proper format. If any statement (except the first one) is incorrect, QUERY issues an error message and prompts you with a ">>". At this point, you may enter one or more statements to replace the statement causing the error. The statements must be on one line and separated by semicolons. If any of these substitute statements is in error, QUERY issues another error message and prompts you again. Once satisfactory statements have been entered, QUERY executes the procedure using the newly-entered statements in place of the statements in error. If the first report statement is in error, the REPORT command terminates.

The corrected statements are not permanent. Once the procedure has executed, the entered statements are lost. The procedure remains unchanged on the Proc-file. To permanently change the procedure, you must use the ALTER command.

Figure 4-4 for example, shows a REPORT procedure named REP4. The third line contains a sort statement with a comma missing. When the procedure is executed using the command:

```
>REPORT REP4
```

```
PROCEDURE: REP4

 001  REPORT
 002  H1,"MONTHLY SHIPMENTS",25,SPACE A2
 003  S1 STOCK#
 004  G1,STOCK#,15
 005  D1,LASTSHIPDATE,25
 006  END
```

Figure 4-4. REPORT Procedure Named REP4

QUERY prints the offending sort statement and prompts for a correction as follows:

```
 003  S1 STOCK#
EXPECTED A ','
>>
```

Once you enter a correct statement or series of statements, execution of the procedure continues. For example, you might respond with:

```
 003  S1 STOCK#
EXPECTED A ','
>>S1,STOCK#;S,LASTSHIPDATE
```

Not only has the sort statement been temporarily fixed, but another sort statement has been added as well. Once the procedure has executed, the corrections and additions are lost.

Statements may also contain options (parameters) which perform such tasks as skipping to the top of the next report page, spacing between report lines, and indicating edit masks to be used to insert punctuation such as decimal points, dollar signs, etc., into values printed in the report. These options are described in table 4-7.

Table 4-7. Statement Parameters

PARAMETER	FUNCTION	APPLICABLE STATEMENTS
print position	determines the rightmost print position (column number) for the *print element*, For character data, this is the rightmost character; for numeric data, it is the position of the least significant digit.	Header, Detail, Group, Total
SPACE A *[number]*	Space *number* lines after printing the report line. If *number* is omitted, one line is spaced.	Header, Detail, Group, Total
SPACE B *[number]*	Space *number* lines before printing the report line. If *number* is omitted, one line is spaced.	Header, Detail, Group, Total
number	is the number of lines to be spaced (from 1 to 5).	
SKIP {A / B}	Skip to the top of the next report page after printing the report line (SKIP A) or before printing the report line (SKIP B).	Detail, Group, Total
E{*number* / Z}	indicates that either an edit mask defined in the identically numbered edit statement (E*number*) is to be used to punctuate a value or, if you use the letter Z, that leading zeros are to be suppressed. In the latter case, no edit statement is required.	Detail, Group Total

SKIPPING AND SPACING

When paging is in effect, the following rules govern skipping and spacing:

1. If a SKIP B and a SPACE B are both associated with the same output line, the SKIP B is processed before the SPACE B.
2. If there are not enough lines remaining on the current page to satisfy a SPACE B, a page is ejected and then the spacing is effected.
3. If a SKIP A and a SPACE A are both associated with the same output line, the SPACE A is ignored.
4. If there are not enough lines remaining on the current page to satisfy a SPACE A, it is treated as a SKIP A.

USING PROCEDURES AND XEQ FILES

SECTION V

A single QUERY command can be fully defined, named, and stored as a procedure in a file for future use. The command forms which may be stored as procedures are listed in table 5-1. A procedure can be defined completely within QUERY using the CREATE and ALTER commands below, or it can be defined by the EDITOR.

Procedures are stored in an MPE ASCII file called a Proc-file. Although you may have more than one file containing procedures, only one Proc-file can be "active" at a time. In this manual, the "active" Proc-file is usually referred to as the current Proc-file.

Before using the Proc-file, you must inform QUERY which one you wish to use through the PROC-FILE= command or in response to the PROC-FILE prompt issued by the DEFINE command. Any procedure command thereafter automatically references the current Proc-file. If a Proc-file name specified in a PROC-FILE= command or in response to the PROC-FILE prompt does not refer to an existing file, QUERY automatically creates a new file for you.

A procedure is executed by referencing it in a FIND, REPORT, or UPDATE *procedure* command. The forms of these commands are specified in Sections III and IV.

Table 5-1. Commands Allowed as Procedures

FIND	UPDATE ADD	UPDATE or U is required.
FIND CHAIN	UPDATE REPLACE	
FIND ALL	UPDATE DELETE	
REPORT		
REPORT ALL		

This section describes the commands you need to work with procedures. The commands and their functions are:

- CREATE — Creates procedures and stores them on the Proc-file indicated in the last PROC-FILE command or prompt. The option CREATE SPACE tells you the number of unused records in the Proc-file.
- DISPLAY — Lists the names of all the procedures stored on the current Proc-file or lists individual procedures complete with line numbers for use in editing.
- ALTER — Edits procedures stored on the current Proc-file, allowing you to delete, replace, or insert new lines into the procedure.
- DESTROY — Removes a procedure from the current Proc-file.
- RENAME — Changes the name of a given procedure.

USING QUERY COMMANDS FROM A FILE

A sequence of QUERY commands can be created and stored using the EDITOR, and subsequently executed using the XEQ command in session or job mode. The XEQ command is described at the end of this section.

CREATE

Stores a command as a named procedure in the currently defined Proc-file.

The form of the CREATE command is

{CREATE / C} *procedure name*, {*filename* / *command*}

For example,

CREATE PROCA, FILEX

procedure name *filename*

or

CREATE PROCZ, FIND BADGE# IS " " END

procedure name *command*

where

procedure name	is a name composed of from 1 to 8 alphanumeric characters. The first character must be alphabetic. No special characters or spaces are allowed. You choose the procedure name. It may not be: ALL, D, DELETE, D*n*, E*n*, END, EZ, G*n*, H*n*, LIST, LP, NOPAGE, PAUSE, R*n*, S, S*n*, SPACE, TF, T*n* where *n* is an integer from 0 to 9.
filename	is the name of an MPE ASCII file containing one of the commands listed in table 5-1. The command will be stored as a procedure in the Proc-file. The *filename* may not be FIND, REPORT, or UPDATE or any identically arranged subset of these characters (such as F, FI, FIN). FINDX and other supersets are acceptable.
command	is one of the commands listed in table 5-1. The form of the command is the same as shown in the command definition in this manual.

You may create a procedure in two ways:

- from input stored in an MPE ASCII file,
- from input entered through the session or job input device.

If the procedure you are creating does not fit in the available Proc-file space, an error message is printed. The incomplete procedure is stored and you may list it with the DISPLAY command.

FILE INPUT

If you create the procedure from a file, QUERY reads the named file when the command is entered. QUERY then lists what is read on the standard list device. and stores it as a procedure on the current Proc-file. The input is copied from the MPE ASCII file to the Proc-file; no attempt is made to check the command for form until the procedure is executed. Blank lines (records) are not copied to the Proc-file.

QUERY does not process the last eight characters of each record in an MPE ASCII file. If you create the file using EDIT/3000 and keep it with numbered lines, the last eight characters are the line numbers. Even if you keep the file unnumbered, QUERY commands or parameters should not be entered in this part of a record.

TERMINAL INPUT

If you enter a procedure through a terminal (in session mode) or through a standard input device (in job mode), the command can be entered on one line or as many lines as necessary without use of the continuation character (&). Whenever you press *return*, QUERY prompts for additional lines by printing >>. To terminate the command, enter a pair of slashes (//) or END as the last three characters in a line. All the characters you enter are stored on the current Proc-file without checking. The procedure is checked for correct form only when it is executed.

QUERY allows a maximum input record of 250 characters (bytes). If you use the continuation character (&), QUERY considers all the lines connected with it as one input record and the total number of characters cannot exceed 250.

CREATE

EXAMPLES

```
>CREATE FPROC, FIND LAST-NAME IS "","","" END
>FIND FPROC
WHAT IS THE VALUE OF - LAST-NAME
>>MURTZ
WHAT IS THE VALUE OF - LAST-NAME
>>FRANZONI
WHAT IS THE VALUE OF - LAST-NAME
>>KENDALL
USING SERIAL READ
Ø  ENTRIES QUALIFIED
>FIND FPROC
WHAT IS THE VALUE OF - LAST-NAME
>>DELLWIG
WHAT IS THE VALUE OF - LAST-NAME
>>MEADOWS
WHAT IS THE VALUE OF - LAST-NAME
>> return
WHAT IS THE VALUE OF - LAST-NAME
>>X
USING SERIAL READ
2  ENTRIES QUALIFIED
```

Create procedure named FPROC. When FPROC is executed, QUERY prompts for three LAST-NAME values

You must supply a value for each null value in the procedure. You can use a known invalid response if you do not want to find the third name.

```
>CREATE CHECK,FIND CREDIT-RATING ILT 5 END
>FIND CHECK
USING SERIAL READ
2  ENTRIES QUALIFIED
>CREATE NAMES,REPORT
>>D,LAST-NAME,2Ø
>>//
>REPORT NAMES
>>PAUSE;
>>END
```

The procedure named CHECK is handy each time you want to list the names of customers with low credit ratings.

The NAMES report procedure can also be used repeatedly.

```
MCFALL
CELERY
```

NOTE

It is sometimes convenient to leave the END out of the procedure. When it is executed, QUERY prompts for the missing END. You can then enter special sort or output control statements to vary the report and its output parameters.

To create a procedure in an MPE ASCII file and enter it in the Proc-file, follow the example below.

Use EDIT/3000 to create the procedure.

```
:EDITOR

HP32201X.4.3B EDIT/3000  WED, JAN 14, 1976, 10:48 AM
/A
    1       FIND
    2       CUSTOMER.ACCOUNT IS ""
    3       END
    4       //
...
/K FIFILE
/E
```

Keep the file.
Exit the Editor.

```
 END OF SUBSYSTEM
:RUN QUERY.PUB.SYS
```

Run QUERY.

```
 HP32216A.03.00 QUERY/3000  WED, JAN 14, 1976, 11:39 AM


 QUERY/3000 READY

>B=STORE
PASSWORD = >>CLERK
MODE = >>1
>PROC-FILE=PROCX
FILE DOES NOT EXIST, BEING CREATED
>CREATE FIPROC,FIFILE
FIND
CUSTOMER.ACCOUNT IS ""
END
>FIND FIPROC
WHAT IS THE VALUE OF - ACCOUNT
>>11111111
NO ENTRY

0  ENTRIES QUALIFIED
```

Define the environment.

Specify a Proc-file.
Use the CREATE command with a procedure name and the file name. (in this example, FIPROC and FIFILE respectively).

CREATE SPACE

Reports the number of unused records left in the current Proc-file.

The form of the CREATE SPACE command is

{CREATE / C} SPACE

For example,

CR SPACE

Each stored procedure starts on a record boundary and extends through as many records as necessary. Generally, one record is sufficient for most UPDATE and FIND commands, while long REPORT commands may take more than one record.

If you want to know how much space is left in the current Proc-file before you begin creating another procedure, you can use the CREATE SPACE command to find out.

EXAMPLE

```
>CREATE SPACE                    The Proc-file has 109 unused records.
RECORDS = 109
>DESTROY SETUP                   If the SETUP procedure is deleted,
>CREATE SPACE                    the Proc-file has 110 unused records.
RECORDS = 110
```

DISPLAY

Lists individual commands stored on the Proc-file as named procedures.

The form of the DISPLAY command when used for this purpose is

$$\left\{\begin{array}{l}\text{DISPLAY}\\ \text{D}\end{array}\right\} \quad \textit{procedure name} \left[\begin{array}{l},m\ [,n]\\ ,\textit{filename}\end{array}\right]$$

For example,

D PROCA, 2, 4
procedure name *m* *n*

or

DISPLAY GETIT
procedure name

or

DISPLAY PROCZ, FILEB
procedure name *filename*

where	*procedure name*	is the name of a command (see table 5-1) stored as a procedure on the current Proc-file.
	m and *n*	are the first and last lines of the procedure (respectively) to be displayed. *m* must be an integer smaller than or equal to *n*. *m* must be an integer greater than or equal to 1.
	filename	is the name of an MPE ASCII file in which you want to write the procedure.

When you use this command, QUERY searches the current Proc-file for the named procedure. If the procedure does not exist or if no Proc-file has been declared, you are so informed and prompted for another command.

If the procedure does exist, it is listed along with line numbers for you to reference when using the ALTER command to edit the procedure. If line numbers (m,n) are included in the DISPLAY command, only those lines specified will be listed.

DISPLAYING TO A FILE

If a *filename* is included in the command, QUERY writes the procedure statements (excluding line numbers and header which appear as part of the procedure listing) to the specified file. The entire procedure must be transferred. Any existing information in the file is overwritten.

DISPLAY

If you have access to the group files and *filename* does not exist in your log-on group, QUERY creates the file and saves it using *filename* as the formal designator (name of the file). The message

FILE DOES NOT EXIST, BEING CREATED

is printed when this situation occurs. The file size will be 200 records. The maximum Proc-file record size is 125 words.

EXAMPLES

```
>DISPLAY FIPROC,FISET
>DISPLAY FIPROC
```

The FIPROC procedure is written to file FISET.

FIPROC is listed on the terminal.

```
PROCEDURE: FIPROC

 001   FIND
 002   CUSTOMER.ACCOUNT IS ""
 003   END
```

If OUTPUT=LP, the list is printed on the QSLIST device. If OUTPUT=TERM, it is printed on the terminal.

```
>O=LP
>DIS F101
>O=TERM
>DIS F101
```

```
PROCEDURE: F101

 001   FIND CHAIN
 002   CUSTOMER.ACCOUNT,SALES.ACCOUNT
 003   IS "" END
```

The procedure lines are numbered.

A single line or a range of lines can be displayed.

```
>DIS F101,2
```

```
PROCEDURE: F101

 002   CUSTOMER.ACCOUNT,SALES.ACCOUNT
 003   IS "" END
```

If a specified file does not exist, QUERY creates one.

```
>DISPLAY F101,SAVEP
FILE DOES NOT EXIST, BEING CREATED
```

DISPLAY LIST

Lists the names of all procedures in the current Proc-file.

The form of the DISPLAY LIST command is

{DISPLAY / D} LIST

For example,

DISPLAY LIST

If you enter this command, QUERY prints the names of all procedures currently in the Proc-file. If no Proc-file has been declared, you are so informed and prompted for another command.

If you have specified OUTPUT=LP (see the OUTPUT= command), the DISPLAY command prints information on both the QSLIST device and the terminal in session mode. In this case, the procedure listing includes a header containing the procedure filename and the date you display the procedure names.

EXAMPLES

```
>DISPLAY LIST

REPORTX      F101         FIPROC       REPORTZ      FNAMES       CHANGE
FPROC        CHECK        NAMES

>P=MANPROC                             If OUTPUT=LP, DISPLAY LISTS lists on
>OUTPUT=LP                             both the terminal and the QSLIST device.
>DISPLAY LIST

F1           LP3          UPD1         REP4         ZAP          BREP


        Listing from QSLIST.
              ↓
PROCEDURE FILE: MANPROC.IMAGE.DATAMGT                    WED, JAN 14, 1976

F1           LP3          UPD1         REP4         ZAP          BREP
```

ALTER

Inserts, replaces, and deletes lines of a procedure stored in the current Proc-file.

The form of the ALTER command is

{ ALTER / A } *procedure name*

For example,

```
ALTER   REP22
        ↑
   procedure name
```

where *procedure name* is the name of a procedure stored on the current Proc-file.

When you enter the ALTER command, QUERY prompts you for insert, replace, and delete statements by printing >>. Each statement operates on a line or range of lines in the procedure. In the statement descriptions that follow, *m* is the first line number in the range and *n* is the last line number. *n* must always be greater than or equal to *m* and *m* must be greater than or equal to 1. Neither *m* nor *n* may exceed the total number of lines in the procedure.

Once you enter the ALTER command, you must do your editing sequentially from lower numbered lines to higher numbered lines. You cannot go back to a line which precedes the ones you are currently editing without exiting the ALTER command and entering another one. This technique is illustrated in the examples which follow.

If an error occurs after a statement is entered, an error message is printed and QUERY prompts you for another ALTER statement.

You can abort the command at any time by entering Control Y (Y^C). If you do this, the procedure remains unchanged. Always terminate the command with /E if you want to save the results of your changes.

The four statements you can use in response to an ALTER prompt are: the insert, replace, delete, and end statements.

Lines which are inserted, or which replace current lines, are not checked for accurate syntax until the procedure is executed. Similarly, if line deletion causes the procedure to have bad command syntax, it will not be detected until the command is executed.

INSERT STATEMENT

The insert statement inserts lines into the procedure.

The form is

>>/I,*m*

>>*line*

>>*line*

.

.

Note: QUERY prompts for each statement and line with >>.

For example,

>>/I,2 ← *m*

>> FIND POTATO = RUSSET END ← *line*

where	*m*	is the number of a procedure line after which you want to insert some statements.
	line	is the new line you want to insert.

The lines which follow the insert statement are inserted into the procedure following line number *m*. You cannot insert lines in front of the first procedure line. For example, the insert statement:

```
>ALTER F1
>>/I,1
>>STOCK# IS "" AND
>>LASTSHIPDATE IS ""
>>/E
```

changes the procedure F1 from:

```
001  FIND
002  END
```

to

```
001  FIND
002  STOCK# IS "" AND
003  LASTSHIPDATE IS ""
004  END
```

To indicate you do not want to insert any more lines, enter another ALTER statement beginning with a slash.

ALTER

REPLACE STATEMENT

The replace statement deletes the line or range of lines specified and replaces them with the lines following the replace statement.

> The form is
>
> >>/R, *m [,n]*
> *(current lines m through n are listed)*
> >> *line*
> >> *line*
> .
> .

Note: QUERY prompts for each statement and line with >>.

For example,

```
>>/R2, 3 <- n
        <- m
HOURS      =  "";
DATE       =  "741022";
>>HOURS    =  "14"; <- line
>>DATE     =  "741101"; <- line
>>STATUS   =  "OK"; <- line
```

where

m	is the number of the first procedure line which you want to replace.
n	is the number of the last procedure line you want to replace. If *n* is not specified, only line *m* is replaced.
line	is the new line you want to replace the existing line.

The example shown above changes the procedure UP1 from:

```
001  UPDATE REPLACE,
002  HOURS="";
003  DATE="741022";
004  END
```

to

```
001  UPDATE REPLACE,
002  HOURS="14";
003  DATE="741101";
004  STATUS="OK";
005  END
```

DELETE STATEMENT

The delete statement deletes the line or range of lines specified.

The form is

>>/D, *m [,n]*

Note: QUERY prompts for each statement with >>.

For example,

```
>>/D, 2,4
      ↑  ↖
      m   n
```

where *m* is the first line you want to delete.

n is the last line you want to delete. If *n* is not specified, only line *m* is deleted. If *n* is specified, the range of lines from *m* to *n* is deleted.

The deleted lines are listed so that you know exactly what you deleted. You cannot delete a procedure from the Proc-file using a delete statement. If you try to delete all the lines in the procedure, an error message is printed. To delete the entire procedure, you must use the DESTROY command.

EXAMPLE

The delete statement:

```
>ALTER REP4
>>/D, 3, 4
S1, STOCK#
S, LASTSHIPDATE
>>/E
```

Note: QUERY lists the deleted lines.

changes the procedure named REP4 from:

```
001  REPORT
002  H1,"MONTHLY SHIPMENTS",25,SPACE A2
003  S1,STOCK#
004  S,LASTSHIPDATE
005  G1,STOCK#,15
006  D1,LASTSHIPDATE,25
007  END
```

to:

```
001  REPORT
002  H1,"MONTHLY SHIPMENTS",25,SPACE A2
003  G1,STOCK#,15
004  D1,LASTSHIPDATE,25
005  END
```

ALTER

END STATEMENT

The end statement terminates the ALTER command. It must always appear as the last statement of the command.

The form is

```
>>/E
```

QUERY continues to prompt you for insert, delete, or replace statements until you enter /E. If the ALTER command is terminated using Control Y the entire command is ignored and the procedure remains in its original state.

EXAMPLE

```
>D REPORTY
```

To alter REPORTY, first display it.

```
PROCEDURE: REPORTY

 001   REPORT
 002   H1,"AS OF:",6
 003   H1,DATE,15
 004   H2,"BOBO'S MERCANTILE",45
 005   H1,"PAGE",69
 006   H1,PAGENO,71
 007   D1,STOCK#,36
 008   D1,LASTSHIPDATE,48,E2
 009   E2,"XX/XX/XX"
 010   END

>ALTER REPORTY
>>/I,9
>>S,LASTSHIPDATE
>>/D,2
INPUT ERROR
>>/E
>ALTER REPORTY
>>/R,2,3
H1,"AS OF:",6
H1,DATE,15
>>H1,"TO DATE:",6
>>H1,DATE,20
>>/D,5,6
H1,"PAGE",69
H1,PAGENO,71
>>/E
>DISPLAY REPORTY
```

Insert a Sort statement after line 9.

You cannot alter a line which precedes the last one altered.

Terminate the command and then you can alter the line.

Replace lines 2 and 3 with new Header statements.

Delete lines 5 and 6.

QUERY lists the deleted lines.

Terminate the command and display the new REPORTY command.

```
PROCEDURE: REPORTY

 001   REPORT
 002   H1,"TO DATE:",6
 003   H1,DATE,20
 004   H2,"BOBO'S MERCANTILE",45
 005   D1,STOCK#,36
 006   D1,LASTSHIPDATE,48,E2
 007   E2,"XX/XX/XX"
 008   S,LASTSHIPDATE
 009   END
```

DESTROY

Deletes a procedure from the current Proc-file.

The form of the DESTROY command is

{ DESTROY / DE } *procedure name*

For example,

```
DESTROY FANG
procedure name
```

where *procedure name* is the name of a command stored on the current Proc-file as a procedure.

This command deletes a procedure from the current Proc-file. If the Proc-file has not been declared, or if the named procedure does not exist on the Proc-file, QUERY so informs you and prompts you for another command. If the procedure does exist on the Proc-file, it is deleted.

EXAMPLE

```
>DISPLAY LIST

F1          F2          UP1          UP2          REP4          ZAP
>DESTROY UP2
>D LIST
```

Delete the UP2 procedure from the current Proc-file.

```
F1          F2          UP1          REP4          ZAP
```

RENAME

Changes the name of a procedure in the current Proc-file.

The form of the RENAME command is

{ RENAME / REN } *old procedure name* , *new procedure name*

For example,

RENAME OPROC , NPROC

old procedure name *new procedure name*

where *old procedure name* is the name currently associated with the procedure.

new procedure name is the name you want to use for the procedure in the future. See the CREATE command for naming rules.

Both procedure names must follow the rules defined in the CREATE command description. An error message is printed if *old procedure name* does not refer to an existing procedure in the current Proc-file.

```
>DISPLAY LIST

F1          F2          UP1          REP4          ZAP

>RENAME UP1,UPD1
>DISPLAY LIST

F1          F2          UPD1         REP4          ZAP

>REN F2,LP
ILLEGAL NAME
>REN F2,LP3
>D LIST

F1          LP3         UPD1         REP4          ZAP
```

Change the name of UP1 to UPD1.

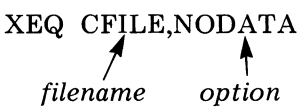
You cannot name a procedure LP, but LP3 is allowed.

Executes QUERY commands from a file instead of the standard input device.

The form of the XEQ command is

$$\left\{\begin{matrix}\text{XEQ}\\ \text{X}\end{matrix}\right\} \quad \textit{filename} \quad [\text{,NODATA}]$$

For example,



where

filename	is the name of an ASCII file containing commands and parameters.
NODATA	is an option. If specified, QUERY prompts for the command parameters when the commands are executed instead of reading them from the file. The prompt consists of two "greater than" symbols.

When the XEQ command is entered, the specified file is read and the commands executed until an end-of-file or another XEQ command is encountered. Any command input is also read from *filename* unless the NODATA option is specified. When an end-of-file is reached, control returns to the original command input device (in session mode, the terminal, or in job mode, the job input device).

Only the first 72 characters of each record are read. You may continue a command on the next record by entering an ampersand (&) as the last non-blank character in the current record.

If an error occurs while opening the data base in session mode, the XEQ file is closed and you are prompted for a QUERY command at the terminal. In job mode, the file is closed and QUERY terminates.

An XEQ command within another XEQ file will close the first file and open the new one. The initial file is never reopened, that is, the XEQ performs as an end-of-file but transfers execution to the new file.

EXAMPLE 1

In this example, an XEQ file called FISET is used to define a particular QUERY session environment.

```
>XEQ FISET

DEFINE
DATA-BASE = STORE
PASSWORD = CLERK
MODE = 5
DATA-SETS = ;
PROC-FILE = PROCX
OUTPUT = TERM
OUTPUT = TERM
END OF XEQ FILE
```

The FISET file contains these records:

DEFINE

STORE

CLERK

5

;

PROCX

TERM

QUERY executes the DEFINE command and uses the other records as data in response to DEFINE command prompts.

EXAMPLE 2

This example illustrates how an XEQ file, PRXFIL, can be used to store and execute a pre-planned find-and-report sequence. It also illustrates how the FIND CHAIN command can be used to access two data sets. Here, information from CUSTOMER and SALES is used to prepare a bill.

```
>XEQ PRXFIL,NODATA

FIND CHAIN CUSTOMER.ACCOUNT,SALES.ACCOUNT="" END
WHAT IS THE VALUE OF - ACCOUNT
>>2
2 ENTRIES QUALIFIED
```

The file PRXFIL looks like this:

```
FIND CHAIN CUSTOMER.ACCOUNT,SALES.ACCOUNT="" END
REPORT BILL
```

Since you want to be prompted for a value when the FIND CHAIN command containing a null value is called, you use the NODATA option in the XEQ command. Note that if REPORT were called directly from the file, XEQ NODATA would prompt you for report statements instead of reading them from the file. Therefore, the REPORT command is stored as a procedure, BILL, in the current PROC-FILE, and is initiated from PRXFIL after the FIND CHAIN command.

EXAMPLE 2 (continued)

The procedure BILL contains the following statements:

```
REPORT
H1,"BOBO'S MERCANTILE",43,SPACE A4
H2,"ACCOUNT #:",11
H2,ACCOUNT,22
H2,"BILLING DATE:",59,SPACE A2
H2,DATE,69
H3,FIRST-NAME,14
H3,INITIAL,16
H3,LAST-NAME,33
H4,STREET-ADDRESS,31
H5,CITY,18
H5,STATE,20
H5,ZIP,27,SPACE A2
H7,"PURCHASE DATE",14
H7,"PURCHASE",65,SPACE B3
H8,"TOTAL",65,SPACE A2
D,PURCH-DATE,10,E1
E1,"XX/XX/XX"
R4,LOAD,QUANTITY
R4,MULT,PRICE
R4,ADD,TAX
D,R4,65,E2
E2,"$$$,$$$.99"
R0,ADD,R4
TF,"TOTAL DUE",15,SPACE B2,SKIP A
TF,R0,65,E2
S, PURCH-DATE
END
```

The resulting report is shown in figure 5-1.

EXAMPLE 2 (continued)

```
                    BOBO'S MERCANTILE


ACCOUNT #:          2                    BILLING DATE:  06/19/79

   HARLEY    W  LOND
    1362 16TH AVE
     SAN FRANCISCO, CA 94122



PURCHASE DATE                                          PURCHASE
                                                          TOTAL

 79/05/05                                              $418.22

     TOTAL DUE                                         $418.22
```

Figure 5-1. Billing Report From an XEQ File.

QUERY/3000 MESSAGES

SECTION VI

All messages to the user are output to the device specified as $STDLIST for the operating system. This is normally a terminal for a session, or the line printer for a job. Messages fall into two categories -- those recognized by QUERY such as incorrect command format, and those passed to QUERY from the IMAGE/3000 routines called by QUERY. Messages in the first category are described in separate tables for each command to which they apply (tables 6-2 through 6-16), or in table 6-1 which contains general messages. IMAGE error messages are described in Appendix A of the IMAGE Reference Manual. In some cases, however, the IMAGE messages are exceptional and may require the aid of HP support personnel if the conditions persist. They are listed in Appendix C of this manual.

Table 6-1. General Messages

MESSAGE	MEANING	ACTION
<CONTROL Y>	You have entered Control Y.	Wait for QUERY prompt.
DATA BASE BUSY, DO YOU WISH TO WAIT (YES OR NO)?	Data base opened with access modes 1 and 5 with other users currently accessing the data base.	Reply NO to ignore current command. Reply YES to be placed in a wait queue until all other users in the queue have accessed the data base.
EOD ON COMMAND INPUT FILE	An :EOD, :JOB, :EOJ, or :DATA record has been encountered.	None.
FATAL ERROR: QUERY TERMINATED	Unrecoverable error has occurred during QUERY operation.	Consult with the data base administrator or the system manager.
FILE OPEN ERROR *code*	File error occurred while MPE opening file on your behalf. *code* is an MPE file system code describing the error.	See the *MPE Commands Reference Manual* for description of the File Information Display which has been printed.
FILE READ ERROR *code*	Physical error during Proc-file read operation. *code* is an MPE file system code.	Same as FILE OPEN ERROR above.
FILE WRITE ERROR *code*	Physical error during file write operation. If *code* = 0, end-of-file was detected.	Same as FILE OPEN ERROR above.
ILLEGAL SOURCE DIGIT IN CONVERSION CVAD – REPLACED WITH ZERO	Routine which converts ASCII numeric strings to packed decimal format has detected an illegal digit. Digit is converted to zero and processing continues.	None.

Table 6-1. General Messages (Continued)

MESSAGE	MEANING	ACTION
INPUT TOO LONG	Command just entered exceeds 698 characters.	Re-enter the command with fewer characters if possible.
INVALID COMMAND	QUERY does not recognize the command.	Consult the command description for proper format and (in session mode) re-enter the command.
PROCEDURE NAME NOT FOUND	Named procedure is not in the directory for the current Proc-file. This message may also appear if the command syntax is incorrect.	Correct the procedure name, declare a different Proc-file, or correct the command syntax.
PROC-FILE BUSY	Proc-file is currently being accessed by another user.	The command using the Proc-file is ignored. Try again later.
PROC-FILE NOT DECLARED	Command has referenced a procedure before a Proc-file has been declared.	Use the PROC-FILE= command to declare a Proc-file.
QUERY/3000 READY	QUERY successfully loaded and ready to accept a command.	Wait for the > prompt. Then enter a command.
READ ERROR FROM COMMAND INPUTFILE	Physical read error occurred while reading from the command input file.	Check the input device and try entering the command again. Consult with MPE system manager.
WRITE ERROR TO COMMAND LISTFILE	Physical write error occurred while writing to the standard list device (known as $STDLIST to MPE).	Check the output device and try the current command again. Consult with MPE system manager.

DEFINE command error messages can be found with other commands. If an error occurs when responding to a DEFINE command prompt, look at the table for the command corresponding to that prompt. For example, the DATA-BASE= ›› prompt messages are included in table 6-4, DATA-BASE= Command Messages.

Table 6-2. ALTER Command Messages

MESSAGE	MEANING	ACTION
END PROCEDURE DATA	Line number referenced in an insert, replace, or delete statement is greater than the final statement number of the procedure being altered.	Command terminates. Re-enter with the correct statement number.
EXPECTED A ','	Comma missing from the alter statement.	Re-enter the statement.
ILLEGAL NAME	Procedure name referenced exceeds eight characters, contains an illegal character, or consists of a reserved word.	Use DISPLAY LIST to determine the correct procedure name. Re-enter command with correct name.
INPUT ERROR	• No line number in an insert, delete, or replace statement (for example, /R,) • Beginning line number greater than ending statement number (for example, /R,5,3) • Beginning line number referenced in statement less than ending line number of previous statement (for example, /R,2,4/D,3) • Alter statement does not start with /I, /R, /D, or /E • Illegal character entered • /R or /I entered with no additional lines (for example, /R,4 /D,6)	Re-enter the statement correctly.
INPUT TOO LONG	Combined length of continued lines (lines connected by & character) has exceeded 250 characters.	Shorten the statement and re-enter the command.
NUMERIC VALUE ERROR	Line number contains too many digits or an illegal character.	Re-enter statement with correct line number.
PROCEDURE CANNOT BE DELETED BY ALTER	You have attempted to delete procedure line by line using ALTER. Procedure is not changed and command terminates.	To delete procedure, use the DESTROY command.

Table 6-2. ALTER Command Messages (Continued)

MESSAGE	MEANING	ACTION
PROC-FILE BUSY, DO YOU WISH TO WAIT (YES OR NO)?	Proc-file being accessed by another user.	Reply NO to abort the command. Reply YES to be placed in wait queue until all users ahead of you in queue have accessed the file.
PROC-FILE OVERFLOW, INPUT TERMINATED	Procedure currently being altered has overflowed the available space left on the Proc-file. Part of procedure may be lost.	List procedure using DISPLAY command before attempting to execute it.
SCRATCH FILE { READ / WRITE } ERROR *code*	Physical read/write error occurred in a QUERY scratch file. File not accessible to users. *code* is an MPE file system code.	See FILE OPEN ERROR in table 6-2. Try command again.

Table 6-3. CREATE Command Messages

MESSAGE	MEANING	ACTION
DIRECTORY OVERFLOW, PROCEDURE REJECTED	Current Proc-file directory is full and cannot accept more procedures.	Delete existing unused procedures on current Proc-file or build and declare another one with PROC-FILE= command.
DUPLICATE PROCEDURE NAME	Procedure name already exists on the current Proc-file.	Create the procedure again with a different name.
EXPECTED A ','	Comma must follow the procedure name.	Re-enter the command.
FILE NOT TYPE ASCII	File referenced is not an ASCII file.	Re-enter the command using the name of an ASCII file.
ILLEGAL NAME	Procedure name exceeds eight characters, contains an illegal character or consists of a reserved word.	See the CREATE command (Section V) for naming rules. Re-enter command with correct name.
INPUT ERROR	Input line contains an illegal character or input line exceeds 250 characters.	Shorten line if possible and re-enter it.

Table 6-3. CREATE Command Messages (Continued)

MESSAGE	MEANING	ACTION
PROC-FILE BUSY, DO YOU WISH TO WAIT (YES OR NO)?	Proc-file being accessed by another user.	Reply NO to abort the command. Reply YES to be placed in wait queue until all users ahead of you in queue have accessed the file.
PROC-FILE OVERFLOW, INPUT TERMINATED	Procedure being entered is too large for remaining space in Proc-file. Part of procedure that fits is stored.	List procedure with DISPLAY command to determine how much was saved.
RECORDS = *xxx*	Message printed in response to CREATE SPACE. *xxx* is number of unused records remaining in Proc-file.	None.

Table 6-4. DATA-BASE= Command Messages

MESSAGE	MEANING	ACTION
BAD DATA BASE REFERENCE	Data base name is invalid or the account or group specified does not exist.	Re-enter command with correct names.
DATA BASE IN USE	Data base is currently being accessed and cannot be opened with exclusive access mode.	Change mode or try later.
DATA BASE OPEN EXCLUSIVELY	Data base is currently being accessed by another user with exclusive access mode.	Wait until other user closes the data base and try again.
DATA BASE OPEN IN ANOTHER MODE	Data base is currently being accessed in an access mode which is incompatible with the one specified.	Try another mode or try again later. (See Section I for QUERY environments/ compatible modes.)
DATA BASE REQUIRES CREATION	Data base root file exists, but files containing data sets do not.	Run the IMAGE/3000 program DBUTIL in CREATE mode to create the data base.
EXPECTED A '='	Command does not include an equal sign (=).	Re-enter the command.
ILLEGAL NAME	Data base name referenced contains an illegal character or more than 24 characters.	Re-enter command with correct name.

Table 6-4. DATA-BASE= Command Messages (Continued)

MESSAGE	MEANING	ACTION
INVALID MODE	Response to the MODE= prompt is not an integer from 1 to 8.	If prompted, re-enter mode. Otherwise re-enter the command.
INSUFFICIENT VIRTUAL MEMORY	QUERY cannot obtain necessary system disc space (virtual memory) to create table and work space required to access the data base.	Try again later.
NO SUCH DATA BASE	Referenced data base does not exist or you did not include the account or group name.	Re-enter the command with the correct data base name or append a group or account and try again.
OUTMODED ROOT FILE	Data base root file was created using a different version of the IMAGE/3000 software, current QUERY software is not compatible with this file.	You may need to dump the data base to tape using the IMAGE/3000 DBUNLOAD program, recreate the data base root file using a different version of DBSCHEMA and DBUTIL, and reload the data base using DBLOAD.
PASSWORD ERROR	Password entered in response to PASSWORD= prompt either contains more than eight characters or does not allow at least read access to one or more data sets in the data base.	If prompted re-enter the password. Otherwise, re-enter the command.
SECURITY VIOLATION	MPE file security has been violated (for example, a user with read access attempts to open a file with write access.)	Try another mode or log-on account or ask the data base administrator for help.
UNABLE TO ACCESS DATA BASE IN THIS MODE, *m, n*	Root file cannot be opened with the access options required for the specific MODE. *m* is the list of required options (AOPTIONS) and *n* is the list of options granted by MPE file system. (*m, n* are integers)	Ask the data base administrator for help.

Table 6-5. DATA-SETS= Command Messages

MESSAGE	MEANING	ACTION
EXPECTED A '='	Command does not include an equal sign (=).	Re-enter the command.
EXPECTED A ','	Data set names in list must be separated by commas.	Re-enter the command.
ILLEGAL DATA SET NAME *name*	Either you cannot access the data set named *name* because your password is not sufficient or *name* does not exist in the data base being accessed.	Declare a different password if possible or use a different data set name.

Table 6-6. DESTROY Command Messages

MESSAGE	MEANING	ACTION
ILLEGAL NAME	Referenced procedure name contains more than eight characters or contains an illegal character or consists of a reserved word.	Use DISPLAY LIST to determine the correct procedure name. Re-enter the command with correct name.
PROC-FILE BUSY, DO YOU WISH TO WAIT (YES OR NO)?	Proc-file is being accessed by some other user.	Reply NO to abort the command. Reply YES to be placed in a wait queue until all users in front of you in the queue have accessed the file.

Table 6-7. DISPLAY Command Messages

MESSAGE	MEANING	ACTION
EXPECTED A ','	Command is missing a comma.	Re-enter the command.
FILE DOES NOT EXIST, BEING CREATED	*filename* specified in the command does not exist in log on group. File is created by QUERY and saved using *filename* as the formal designator. File size is 200 records.	None.
ILLEGAL NAME	Referenced procedure name contains more than eight characters, contains an illegal character, or consists of a reserved word.	Use DISPLAY LIST to determine the correct procedure name. Re-enter the command with correct name.

Table 6-7. DISPLAY Command Messages (Continued)

MESSAGE	MEANING	ACTION
INPUT ERROR	• The beginning line number is less than 1 (for example, DISPLAY F101,0,3) • The beginning line number is greater than the ending line number (for example, DISPLAY F101,4,2)	Re-enter the command.
NUMERIC VALUE ERROR	A line number contains an illegal character or too many characters (for example, DISPLAY F101,2A,3)	Re-enter the command.
PROC-FILE BUSY, DO YOU WISH TO WAIT (YES OR NO)?	Proc-file is being accessed by another user.	Reply NO to abort the command. Reply YES to be placed in a wait queue until all users ahead of you in queue have accessed the file.
SCRATCH FILE WRITE ERROR *code*	Physical write error has occurred in a QUERY scratch file. File is not accessible to users. *code* is an MPE file system code.	See FILE OPEN ERROR in table 6-2. Try command again.

Table 6-8. FIND Command Messages

MESSAGE	MEANING	ACTION
BROKEN CHAIN POINTERS	You have read only access and are sharing data base with user who is making structural changes to it.	Re-enter the command.
COMMAND TABLE OVERFLOW	FIND command contains more than 50 logical relationships.	Re-enter command with fewer relationships.
DATA BASE NOT DECLARED	FIND command has been entered prior to declaring the data base to be accessed.	Use the DEFINE or DATA-BASE= command to declare the data base.
DATA ITEM VALUE TOO LONG	Data item value entered exceeds 500 characters or the maximum allowable length for the data item type defined.	Re-enter command with a different value.
ENTRIES RETRIEVED EXCEEDS 65000	Only 65000 entries can be selected as a result of a single FIND command.	Re-enter command with stricter selection criteria or use the entries already selected.

Table 6-8. FIND Command Messages (Continued)

MESSAGE	MEANING	ACTION
xxx ENTRIES QUALIFIED	*xxx* is the number of entries selected by the FIND command.	None.
EXPECTED A CONNECTOR OR 'END'	Command is missing a logical connector (AND or OR) or a terminating "END", or an ending quote is missing from a data item value.	Re-enter command.
EXPECTED A LITERAL VALUE	Beginning or ending quote (") is missing from a data item value.	Re-enter the command.
'FIND' EXPECTED	Procedure called from the Proc-file through a FIND *procedure name* command is not a proper FIND procedure.	Use DISPLAY or DISPLAY LIST to determine the problem and re-enter the command with the correct procedure name.
ILLEGAL DATA ITEM NAME *name*	Data item *name* does not belong to the data set specified or to the data base currently being accessed.	Re-enter the command with a valid data item name. (Use the FORM command to determine the valid data item name.)
ILLEGAL DATA SET NAME *name*	Data set called *name* does not belong to the data base currently being accessed.	Re-enter the command with a valid data set name. (Use the FORM command to determine the valid set names.)
ILLEGAL ITEM LENGTH	Data item being referenced is not of the proper size and type to be processed by QUERY.	See Section I for description of acceptable data types.
INPUT TOO LONG – TRUNCATED	Data item value entered is longer than the defined length of the data item. Value is truncated on the left and stored.	None.
INVALID CONNECTOR OR TERMINATOR *xxxx*	QUERY expected a logical connector (AND or OR) or "END". *xxxx* is the offending character string appearing in place of the expected string.	Re-enter the command.
INVALID # VALUES FOR RELATIONAL OPERATOR	Multiple data item values may follow only the "equal" or "not equal" relational operators.	Re-enter the command changing either the relational operator or values.

Table 6-8. FIND Command Messages (Continued)

MESSAGE	MEANING	ACTION
INVALID RELATIONAL OPERATOR	A relational operator other than the acceptable ones shown in table 3-1 has been entered.	Re-enter command after consulting table 3-1 and replacing the incorrect operator.
NO ENTRY	No entries were located which match the criteria specified in the FIND command.	Enter next command.
NON-NUMERIC IN REAL VALUE	Non-numeric digit has appeared in a real data item value. (This does not include values entered in scientific notation such as 12.44E+04.)	Re-enter command with correct value.
NOT ENOUGH SECTORS IN QSKIB FILE	More data item values appear in a FIND command than can be stored in a job temporary file known to QUERY as QSKIB.	Re-enter the command with fewer data item values.
PROCEDURE NAME TOO LONG	Name of the procedure appearing in a FIND *procedure name* command is more than eight characters.	Re-enter the command with a valid procedure name.
RETRIEVAL FROM MORE THAN ONE DATA SET	FIND command has specified access to more than one data set or a FIND CHAIN command has specified access to more than one detail data set.	Determine the problem by using the FORM command and examining the FIND command just entered. Re-enter the command.
SCRATCH FILE {READ / WRITE} ERROR *code*	Physical error has occurred in a QUERY scratch file. File is not accessible to users. *code* is an MPE file system code.	See FILE OPEN ERROR in table 6-2. Try command again.
USING SERIAL READ	QUERY must read each record of the detail data set without benefit of a master data set.	Wait for QUERY to complete the search or terminate the command with Control Y.
WHAT IS THE VALUE OF *name*	FIND command contains a null data item value.	Supply a value for the data item called *name*.

Table 6-9. FORM Command Messages

MESSAGE	MEANING	ACTION
DATA BASE NOT DECLARED	FORM command has been entered prior to declaring the data base to be accessed.	Use the DEFINE or DATA-BASE= command to declare the data base name.
ILLEGAL DATA ITEM NAME *name*	Data item named *name* does not appear in the data base being accessed or is not accessible to you based upon your password.	Use the FORM command to determine the correct item or change your password with the PASSWORD= command.
ILLEGAL DATA SET NAME *name*	Data set called *name* does not belong to the data base currently being accessed or is not accessible to you based upon your password.	Same as previous message.

Table 6-10. LIST Command Messages

MESSAGE	MEANING	ACTION
COMMAND TABLE OVERFLOW	LIST command contains more than 10 logical relationships.	Re-enter the command with fewer logical relationships.
CONSTANT LITERAL TABLE OVERFLOW	Number of data item values appearing in the entire LIST command has exceeded the capacity of the table used to hold them.	Re-enter the command with fewer data item values.
DATA BASE NOT DECLARED	LIST command has been entered prior to declaring the data base to be accessed.	Use the DEFINE or DATA-BASE= command to declare the data base.
DATA ITEM VALUE TOO LONG	Data item value entered exceeds 500 characters or the maximum allowable length for the data item type defined.	Re-enter command with a different value.
DUPLICATE ITEM IGNORED *name*	Same data item has appeared more than once in the data item list. The first occurrence of the name is used and others are ignored.	None.

Table 6-10. LIST Command Messages (Continued)

MESSAGE	MEANING	ACTION
EXPECTED A CONNECTOR OR 'END'	Command is missing a logical connector (AND or OR) or a terminating "END" or a data item value is missing a terminating quote.	Re-enter the command.
ILLEGAL DATA ITEM NAME *name*	Data item *name* does not belong to the data set specified or to the data base currently being accessed.	Re-enter the command with a valid data item name. (Use the FORM command to determine the valid data item name.)
ILLEGAL DATA SET NAME *name*	Data set called *name* does not belong to the data base currently being accessed.	Re-enter the command with a valid data set name. (Use the FORM command to determine the valid set names.)
ILLEGAL ITEM LENGTH	Data item being referenced is not of the proper size and type to be processed by QUERY.	See Section I for description of acceptable data types.
ILLEGAL ITEM LENGTH – ITEM IGNORED	Item length exceeds the maximum for the data type and is not printed. Item maximums are: U136, X136, Z20, and P20.	See table 1-1 for maximum characters for each data type.
INPUT TOO LONG – TRUNCATED	Data item value entered is longer than the defined length of the data item. Value is truncated on the left for numeric type items and on the right for U and X type values.	None.
INVALID CONNECTOR OR TERMINATOR *xxxx*	QUERY expected a logical connector (AND or OR) or "END". *xxxx* is the offending character string appearing in place of the expected string.	Re-enter the command.
INVALID NUMERIC DIGIT	Value input for a data item defined as type Z or P contains an invalid or non-numeric digit.	Re-enter the command with the correct value.
INVALID RELATIONAL OPERATOR	A relational operator other than the acceptable ones shown in table 3-1 has been entered.	Re-enter the command after consulting table 3-1 and replacing the incorrect operator.

Table 6-10. LIST Command Messages (Continued)

MESSAGE	MEANING	ACTION
INVALID # VALUES FOR RELATIONAL OPERATOR	Multiple data item values may follow only the "equal" or "not equal" relational operators.	Re-enter the command changing either the relational operator or values.
NON-NUMERIC IN REAL VALUE	Non-numeric digit has appeared in a real data item value. (This does not include values entered in scientific notation such as 12.44E+04.)	Re-enter command with correct value.
NUMERIC VALUE ERROR	Numeric value input contains an illegal digit or the value exceeds the maximum value allowable for the data type.	See table 1-1 description of data item types. Re-enter command with correct value.
RETRIEVAL FROM MORE THAN ONE DATA SET	LIST command has specified access to more than one data set.	Use the FORM command to determine the data base structure and examine the LIST command just entered. Re-enter the command.
WHAT IS THE VALUE OF *name*	LIST command contains a null data item value.	Supply a value for the data item called *name.*

Table 6-11. OUTPUT= Command Messages

MESSAGE	MEANING	ACTION
EXPECTED A '='	Equal sign is missing from the command.	Re-enter the command.
FILE NOT TYPE ASCII	File equated to the QSLIST file (through an MPE :FILE command) is not an ASCII file.	Check the file and re-enter the command.
ILLEGAL NAME	Name following OUTPUT= is not LP or TERM.	Re-enter command.
INVALID COMMAND	OUTPUT=LP has been entered twice without an intervening OUTPUT= TERM. Command is ignored the second time it is entered.	None.
'QSLIST' DEVICE NOT AVAILABLE	Device defined as QSLIST is in use by another process.	Wait and try again later or use the terminal.

Table 6-12. PROC-FILE= Command Messages

MESSAGE	MEANING	ACTION
EXPECTED A '='	Equal sign (=) is missing from the command.	Re-enter the command.
FILE DOES NOT EXIST, BEING CREATED	*filename* specified does not exist in the log-on group. File is created by QUERY and saved using *filename* as the formal designator. File size is 200 records.	None.
FILE NOT TYPE ASCII	File referenced is not an ASCII file.	Re-enter the command using the name of an ASCII file.
FILE OPEN ERROR BAD RECSIZE	ASCII file being declared as the Proc-file does not contain a record size of 256 bytes.	Declare a different Proc-file with the PROC-FILE= command.
ILLEGAL NAME	Proc-file name contains an illegal character.	Re-enter command with valid name.
OLD FORMAT ON PROCEDURE FILE	File specified as the Proc-file does not have a file code of 1070.	Ask the data base administrator for help.
PROC-FILE TOO LARGE	File being declared as Proc-file is larger than 400 records.	Re-enter command with a different file name.
PROC-FILE TOO SMALL	File being declared as the Proc-file is smaller than five records.	Re-enter command with a different file name.
UNABLE TO USE FILE	The requested new Proc-file *name* is not large enough or is not an empty existing Proc-file.	Re-enter command with a different *name.*

Table 6-13. RENAME Command Messages

MESSAGE	MEANING	ACTION
DUPLICATE PROCEDURE NAME	New procedure name already exists on the current Proc-file.	Re-enter command with a different procedure name.
EXPECTED A ','	Comma must follow the old procedure name.	Re-enter the command.
ILLEGAL NAME	Referenced procedure name contains more than eight characters, contains an illegal character, or consists of a reserved word.	See the CREATE command (Section V) for naming rules. Re-enter statement with correct name.
PROC-FILE BUSY, DO YOU WISH TO WAIT (YES OR NO)?	Proc-file being accessed by another user.	Reply NO to abort the command. Reply YES to be placed in wait queue until all users ahead of you in queue have accessed the file.

Note: For messages in table 6-14, if the statement in error is the first report statement, the command terminates. If it is not the first statement, you can re-enter the statement which caused the error as directed in the ACTION column. If the REPORT is a procedure, you must correct the statements later using ALTER if you want them to be permanently changed.

Table 6-14. REPORT Command Messages

MESSAGE	MEANING	ACTION
ADD OPTION ALREADY SPECIFIED	The ADD option has been entered twice for the same total statement.	Re-enter statement with a single ADD option.
ADD OPTION NOT ALLOWED IN STATEMENT	Statement other than total contains an ADD option.	Re-enter statement without ADD.
ALPHA EDIT MASK EXCEEDS MAXIMUM SIZE	Number of characters in an alphanumeric edit mask exceeds the length of the output device record.	Correct the edit mask length by changing the edit statement.
ARITHMETIC OVERFLOW [ON R*n*]	Sum of data item values being totalled (using a total statement ADD option) exceeds 20 digits. A register arithmetic operation caused an underflow/overflow condition on the specified register number *n*.	REPORT command terminates. Re-enter the command or alter the report procedure and execute it again.

Table 6-14. REPORT Command Messages (Continued)

MESSAGE	MEANING	ACTION
AVERAGE OPTION ALREADY SPECIFIED	The AVERAGE option has been entered twice for the same total statement.	Re-enter the statement with a single AVERAGE option.
AVERAGE OPTION NOT ALLOWED IN STATEMENT	Statement other than total contains an AVERAGE option.	Re-enter the statement without AVERAGE.
COMMAND TABLE OVERFLOW	REPORT command contains more than 100 statements.	Re-enter report with fewer statements.
CONSTANT LITERAL TABLE OVERFLOW	Number of literal character strings appearing in all the header, detail, group, and total statements have exceeded the capacity of the table used to hold them.	Revise the report and re-enter the command.
CONTROL BREAK INCONSISTENCY	Level numbers of total or group statements do not match properly with sort statement level numbers.	Revise the report and re-enter the command.
COUNT OPTION ALREADY SPECIFIED	The COUNT option has been entered twice for the same total statement.	Re-enter the statement with a single COUNT option.
COUNT OPTION NOT ALLOWED IN STATEMENT	Statement other than total contains a COUNT option.	Re-enter the statement without COUNT.
DATA ITEM NOT RETRIEVED	Statement has referenced a data item that is not a member of the data entries selected by the last FIND command.	Revise the statement and re-enter the command or statement or use the appropriate FIND command and re-enter the REPORT command.
DUPLICATE EDIT STATEMENTS	More than one edit statement with the same number has been entered.	The second edit statement is ignored unless the REPORT is a procedure. If so, reply *return* to prompt. Alter the procedure later.
DUPLICATE SORT DATA ITEM NAME	More than one sort statement contains the same data item name.	Revise report and re-enter the command.
EDIT MASK ERROR	A numeric edit mask contains characters in the wrong order.	Correct the mask and re-enter the edit statement.
EDIT OPTION ALREADY SPECIFIED	Statement contains more than one edit statement reference.	Revise statement and re-enter it.

Table 6-14. REPORT Command Messages (Continued)

MESSAGE	MEANING	ACTION
EDIT OPTION ERROR	Edit statement reference is not between E0 and E9 inclusive.	Revise statement and re-enter it.
EDIT OPTION NOT ALLOWED IN STATEMENT	Edit statement has been referenced in a statement which contains a literal as a *print element.*	Revise report and re-enter command.
END-OF-FILE ON REPORT FILE	There is not enough room left on the file to which the report is being written.	Notify the system operator.
END PRINT POSITION ERROR	The end print position specified contains an illegal digit, is equal to zero, or is greater than the record size of the output device.	Revise the statement and re-enter it.
EXPECTED A LITERAL VALUE	QUERY expected to find a literal character string in the statement.	Revise the statement and re-enter it.
EXPECTED A ","	Comma is missing from a statement.	Re-enter the statement.
EXPECTED A ";"	More than one statement appears on the same line without being separated by a semicolon.	Re-enter the statements on separate lines or separated by semi-colons.
HEADERS OVERFLOW A PAGE	Header statements print more header lines of information (counting line skipping) than can fit on one output page.	Revise the header statements and re-enter the command.
ILLEGAL ASCENDING/ DESCENDING CODE	ASC or DES sort parameter is missing or incorrectly spelled.	Revise sort statement and re-enter it.
ILLEGAL DATA ITEM NAME *name*	Data item named *name* does not belong to data base currently being accessed, is not accessible to you based upon your password, or is not a member or the data set specified in the fully-qualified data item name.	Use a different data item name or change your password by using the PASSWORD= command.
ILLEGAL DATA SET NAME *name*	Data set called *name* does not belong to data base currently being accessed or is not accessible to you based upon your password.	Use a different data set name or change your password with the PASSWORD= command. Re-enter the command.

Table 6-14. REPORT Command Messages (Continued)

MESSAGE	MEANING	ACTION
ILLEGAL DATA TYPE *name*	Data type specified in a register statement is defined as U or X or is not R*n* where *n* is a number from 0 to 9. *name* is the data item name.	Revise the register statement and re-enter it.
ILLEGAL ITEM LENGTH	Data item being referenced is not of the proper size and type to be processed by QUERY.	See Section I for description of acceptable data types.
ILLEGAL NAME	Name of a REPORT procedure exceeds eight characters or does not start with an alphabetic character.	Re-enter command with correct name.
ILLEGAL REGISTER OPERATOR	Operator specified is not one of the valid register operators.	See the register statement description in Section IV for a list of valid operators. Revise statement and re-enter it.
INCONSISTENCY BETWEEN OPTION AND EDIT STATEMENTS	Statement contains an edit statement label not used to define an edit statement.	Revise statement and re-enter command.
INVALID NUMERIC DIGIT	A numeric value specified an illegal digit or the value exceeds the maximum value.	Re-enter the statement.
LEVEL ERROR	Level number appearing in sort, group, or total statement is less than 1, greater than 5, or a header or detail statement level number is not between 1 and 9 inclusive.	Revise necessary statement and re-enter it.
LITERAL TOO LARGE	Literal character string exceeds the record size of the output device.	Revise statement containing string and re-enter it.
MORE THAN 5 FIELDS ARE BEING TOTALED	Total statements reference more than five different data items.	Revise necessary total statements and re-enter command.
NON-NUMERIC IN REAL VALUE	A non-numeric digit appears in the real value or exceeds the maximum allowable value. (Does not include values entered in scientific notation such as 12.44E+04.)	Re-enter the statement.

Table 6-14. REPORT Command Messages (Continued)

MESSAGE	MEANING	ACTION
NO. OF SORT ITEMS EXCEEDS LIMIT	Number of sort items defined exceeds the maximum of 66.	Revise report and re-enter command.
NOT A REPORT PROCEDURE	Procedure referenced in REPORT *procedure name* command is not a proper REPORT procedure.	Use DISPLAY or DISPLAY LIST to find the correct procedure name.
NOT A REPORT STATEMENT	Statement is not recognized as a valid report statement.	Revise statement and re-enter it.
NUMERIC EDIT MASK EXCEEDS 20 CHARACTERS	Numeric edit mask in an edit statement exceeds the maximum allowable 20 characters.	Revise the edit mask and re-enter the statement.
OPTION ERROR	Statement option is either illegal for the statement in which it appears, or the option is not recognized as valid.	Revise statement and re-enter it.
PARAMETER ERROR	The output control statement OUTPUT=LP has a bad format.	Revise statement and re-enter command.
RECORD HAS NOT YET BEEN FOUND	REPORT command has been executed before a FIND command has been entered to select entries for reporting.	Enter FIND command and re-enter REPORT command.
REPORT CANNOT BE GENERATED DUE TO ERRORS	REPORT command contains errors and will not generate a report.	Revise report and re-enter command.
REPORT INCOMPLETE – ITEM MISSING	Indicates that some of the data items selected by the FIND command may have been deleted by some other user prior to executing the REPORT command.	Re-enter the FIND and REPORT commands.
SAME LINES HAVE CONFLICTING REPORT OPTIONS *label*	Like statements (such as multiple statements labeled D2) contain the same statement option (such as SKIP A). A statement option may appear only once for statements concerning the same report line. *label* is the statement type and level.	Revise necessary statements and re-enter command.

Table 6-14. REPORT Command Messages (Continued)

MESSAGE	MEANING	ACTION
SCRATCH FILE {READ / OPEN / WRITE} ERROR *code*	Physical error has occurred during QUERY access of a scratch file. File is not accessible to user. *code* is an MPE file system code.	See FILE OPEN ERROR in table 6-2. Try command again.
SKIP OPTION ALREADY SPECIFIED	Skip option has been specified twice in the same statement.	Revise statement and re-enter command.
SKIP OPTION ERROR	Skip option is not SKIP A or SKIP B.	Revise statement and re-enter command.
SKIP OPTION NOT ALLOWED IN HEADER	Header statements may not contain SKIP options.	Revise statement and re-enter command.
SORTLIB: *message*	Error has occurred in one of the SORT/3000 procedures operating in behalf of QUERY. *message* is the SORT/3000 error message as specified in the *SORT/3000 Reference Manual.*	Consult the SORT/3000 manual and try the command again.
SORT LEVEL MISSING OR DUPLICATE *label*	Group or total statements reference sort statements which either do not exist or appear twice in the REPORT command. For example, S1 appears twice, or a total statement is labeled T3 when no S3 statement exists. *label* is the statement type and level.	Revise necessary statements and re-enter command.
SPACE OPTION ALREADY SPECIFIED	A SPACE A or SPACE B option appears twice in the same statement.	Revise statement and re-enter command.
SPACE OPTION ERROR	SPACE option is not SPACE A *n* or SPACE B *n*, where *n* is 1 to 5 inclusive.	Revise statement and re-enter command.
STATUS=%XXXXXX P=%XXXXXX TRAP=%XXXXXX	Hardware trap has occurred which was not enabled by QUERY and cannot be handled by QUERY. May be caused by hardware failure. QUERY terminates. XXXXXX is an octal number. Status is the status register, P is the P-register and TRAP is the MPE trap number.	Consult with the data base administrator or system manager.

Table 6-15. UPDATE Command Messages

MESSAGE	MEANING	ACTION
ATTEMPTED DELETION OF CHAIN HEAD	Attempt has been made to delete a master search item value that still exists as a detail search item value in one or more linked detail data sets.	If desired, delete detail entries first and then master entry.
ATTEMPTED MODIFICATION OF A CRITICAL ITEM	UPDATE REPLACE command has attempted to change the value of a data item defined as a search or sort item.	Enter next command or delete entire entry and add again with new value for search or sort item.
ATTEMPTED MODIFICATION OF A READ-ONLY ITEM	UPDATE REPLACE command has attempted to modify a data item to which you have only read but not write access.	You cannot modify the item unless you supply the appropriate password with the PASSWORD= command.
AUTOMATIC MASTER	UPDATE ADD command has attempted to add an entry to an automatic master data set.	Enter another command.
BUFFER OVERFLOW	Total length of all data items to be modified in data set exceeds 2048 words and has overflowed the buffer used to hold them. If UPDATE REPLACE, the command terminates.	Shorten command and re-enter it.
DATA ITEM NOT RETRIEVED	UPDATE REPLACE command entered which references a data item that was not previously selected by a FIND command.	Use FIND command and re-enter UPDATE command.
DATA SET FULL	No more entries can be added to the data set.	Consult with the data base administrator.
DATA SET NOT WRITEABLE	Command entered attempting to modify a data item in a data set which you are only allowed to read but not write.	You must supply an appropriate password or enter a different command.
DELETE ALL RETRIEVED ENTRIES (YES OR NO)?	Asks for confirmation of UPDATE DELETE command just entered.	Reply YES to delete all entries selected by the most recent FIND command. Reply NO to abort the command.

Table 6-15. UPDATE Command Messages (Continued)

MESSAGE	MEANING	ACTION
DUPLICATE ITEM IGNORED *name*	Same data item has appeared more than once in an UPDATE REPLACE command. Only first occurrence is used.	None.
EXPECTED A LITERAL VALUE	Value in quotes following data item name is missing in an UPDATE REPLACE command.	If prompted with >>, re-enter the data item name and value. Otherwise, re-enter the entire command.
EXPECTED A "="	Equal sign (=) missing from an UPDATE REPLACE command.	Re-enter command.
EXPECTED A ","	Comma is missing after UPDATE ADD and before data set name.	Re-enter command.
EXPECTED A ";"	Multiple data item name/data item value pairs exist in an UPDATE REPLACE command and are not separated by semicolons.	Re-enter command.
FULL CHAIN FOR *item/set*	Attempt has been made to add a detail entry to a chain which has reached the maximum allowable number of entries (65535). Current attempt is ignored. *item* is detail data set search item name and *set* is master data set name.	Consult with data base administrator.
FULL MASTER FOR *item/set*	Attempt has been made to add a detail data entry with a search item value that does not match any existing search item value in the corresponding automatic master data set (*set*), and a new master entry cannot be created because the automatic master data set is full. *item* is the detail data set search item name.	Consult with the data base administrator.
ILLEGAL ACCESS	You do not have write access to the data base because your access mode does not allow it.	See Section I for a description of access modes and capabilities associated with each.

Table 6-15. UPDATE Command Messages (Continued)

MESSAGE	MEANING	ACTION
ILLEGAL DATA ITEM NAME *name*	Data item called *name* either does not appear in the data base being accessed or is not a member of the specified data set.	Re-enter command with correct data item name. Use FORM command to determine valid data item name.
ILLEGAL DATA SET NAME *name*	Data set called *name* is not a member of the data base currently being accessed.	Check the data set name by using the FORM command and change the data base name if necessary.
ILLEGAL ITEM LENGTH – ITEM IGNORED	Data item being referenced is not the proper size and type to be processed by QUERY. If the data item is a search or sort item, the command terminates.	See Section I for a discussion of data types supported by QUERY.
ILLEGAL NAME	UPDATE procedure name exceeds eight characters, contains an illegal character, or consists of a reserved word.	Use DISPLAY LIST to list the procedures in the Proc-file.
ILLEGAL UPDATE TYPE	Character string following UPDATE is not ADD, REPLACE, or DELETE.	Re-enter the command.
INPUT ERROR – REENTER	Illegal character has been input and line is ignored.	Re-enter the command or line.
INPUT TOO LONG – TRUNCATED	Data item value entered is longer than the maximum defined length for the data item. Value is truncated on the right and processing continues.	None.
INVALID NUMERIC DIGIT	Value entered for a data item defined as type Z or P contains an invalid or non-numeric digit.	Check value and re-enter command.
MISSING CHAIN HEAD FOR *item/set*	Attempt made to add an entry to a detail data set linked to the specified master data set without a corresponding entry in the master data set for the search item value. *item* is the detail data set search item and *set* is the master data set name.	Add an entry with the same search item value to the master data set and then re-enter this command.

Table 6-15. UPDATE Command Messages (Continued)

MESSAGE	MEANING	ACTION
NON-NUMERIC IN REAL VALUE	Value entered for a data item defined as type real contains an illegal character, or the value is larger than the maximum allowable value for the data type.	See Section I for a discussion of data types and ranges. Re-enter command with correct value.
NOT AN UPDATE PROCEDURE	Referenced procedure is not a proper UPDATE procedure.	Use DISPLAY and DISPLAY LIST to determine correct procedure name. Re-enter command.
NUMERIC VALUE ERROR – ITEM IGNORED	Value entered contains an illegal character or too many characters.	Check value and re-enter command.
REAL VALUE ERROR – ITEM IGNORED	Real value contains an illegal character or is not in the proper range for the data item defined.	See Section I for a discussion of data types and ranges. Re-enter command with correct value.
RECORD HAS NOT YET BEEN FOUND	UPDATE REPLACE or UPDATE DELETE command has been entered prior to selecting entries with a FIND command.	Use the FIND command and then re-enter the UPDATE command.
SCRATCH FILE READ ERROR *code*	Physical error has occurred while MPE file system reading from a QUERY scratch file. File is not accessible to user. *code* is an MPE file system code.	See FILE OPEN ERROR in table 6-2. Try command again or consult with data base administrator or system manager.
UPDATE TO MORE THAN ONE SET	UPDATE {DELETE / REPLACE} command used on entries selected with FIND CHAIN command.	You can only use the FIND command to select entries to be replaced or deleted.
WHAT IS THE VALUE OF *name*	Prompt for a data item value. *name* is the name of the data item in the UPDATE command that contains a null data item value.	Supply the requested data item value.

Table 6-16. XEQ Command Messages

MESSAGE	MEANING	ACTION
END OF XEQ FILE	End-of-file encountered in XEQ file.	None.
EXPECTED A ","	Comma must follow the file-name when using the optional NODATA parameter.	Re-enter command with comma.
FILE NOT TYPE ASCII	The XEQ file is not an ASCII file.	Check the file and re-enter the command with the correct file type.
ILLEGAL NAME	XEQ filename is not valid (begins with a non-alphabetic character.)	Check the name and the *MPE Commands Reference Manual* for valid file names.
PARAMETER ERROR	Optional parameter NODATA is not recognizable.	Re-enter command with parameter spelled properly.

QUERY/3000 COMMAND SUMMARY

APPENDIX A

ADD	see UPDATE ADD
ALTER	Inserts, replaces, and deletes lines of a procedure stored in the Proc-file. {ALTER / A} *procedure name* /I,*m* /R,*m*[,*n*] /D,*m*[,*n*] /E *m* and *n* are line numbers; $m < n$
CREATE	Creates new FIND, REPORT, or UPDATE procedures. {CREATE / C} SPACE or {CREATE / C} *procedure name*, {*filename* / *command*}
DATA-BASE=	Indicates which data base is to be accessed. {DATA-BASE= / B=} *data base name*
DATA-SETS=	Indicates which data sets to access. {DATA-SETS= / S=} *[data set list]*
DEFINE	Lists the status of all Environment commands and enables changes. {DEFINE / DEF}
DELETE	see UPDATE DELETE
DESTROY	Deletes a procedure from the Proc-file. {DESTROY / DE} *procedure name*

DISPLAY

Lists a procedure or lists the names of all the procedures stored in the Proc-file.

{DISPLAY / D} LIST

or

{DISPLAY / D} *procedure name* [*,m[,n]* / *,filename*]

EXIT

Terminates QUERY execution.

{EXIT / E}

FIND

Locates data entries in the data base according to user specifications.

{FIND / F} *procedure name [,character]*

or

{FIND / F} *relation* [{AND / OR} *relation....*] [END]

or

{FIND / F} CHAIN *item identifier* {IS / IE / EQ / =} *"value"* [{AND / OR}.....][END]

or

{FIND / F} ALL *[data set name.] data item name*

FORM

Lists information about the structure of the data base currently being accessed.

{FORM / FO} [*data set name* / *data item name* / SETS / ITEMS / PATHS]

HELP

Prints information about the QUERY command set.

{HELP / H} [*command name* [FUNCTION / FU] [FORMAT / FO] [PARAMETERS / PA]]

LIST	Prints complete or partial entries from a single data set with automatic formatting and headings. {LIST / L} {*data set name* / *data item list*} [FOR *relation* [{AND / OR} *relation...*]] [END]
MODE=	Changes the mode of access. {MODE= / M=} *mode number*
OUTPUT=	Specifies the QUERY list output device for some of the QUERY commands. {OUTPUT= / O=} {TERM / LP}
PASSWORD=	Changes your password. {PASSWORD= / PA=} *password*
PROC-FILE=	Specifies which file QUERY is to use to store and execute procedures. {PROC-FILE= / P=} *filename [,n]*
RENAME	Changes a procedure name. {RENAME / REN} *old procedure name, new procedure name*
REPLACE	see UPDATE REPLACE.
REPORT	Lists the item values located by a FIND command. {REPORT / R} *[output control statements]* ALL [,character] or {REPORT / R} *report statements* END or {REPORT / R} *[output control statements] procedure name [,character]*

REPORT Statements:

Header

H *header number, print element, print position* [,SPACE A *[number]*]

[,SPACE B *[number]*]

Detail

D *[detail number] ,print element, print position* [,SPACE A *[number]*]

[,SPACE B *[number]*] [,SKIP {A / B}] [,E {*number* / Z}]

Edit

E *number, "edit mask"*

Sort

S *[level] ,data item name* [, {ASC / DES}]

Group

G *level, print element, print position* [,SPACE A *[number]*]

[,SPACE B *[number]*] [,SKIP {A / B}] [,E {*number* / Z}]

Total

T *level, print element, print position* [,SPACE A *[number]*]

[,SPACE B *[number]*] [,SKIP {A / B}] [,E {*number* / Z}]

[, {ADD / AVERAGE / COUNT}]

Register

R *number*, {L[OAD] / A[DD] / S[UBTRACT] / M[ULTIPLY] / D[IVIDE]} , *data element*

Output Control

LINES =*integer*

NOPAGE

[OUT=] LP

PAUSE

UPDATE

Executes an UPDATE command procedure.

{UPDATE / U} *procedure name [,character]*

UPDATE ADD	Adds data entries to the data base. { UPDATE ADD, U ADD, ADD [,] AD [,] } *data set name*
UPDATE DELETE	Deletes data entries from the data base. { UPDATE DELETE U DELETE DELETE DEL }
UPDATE REPLACE	Modifies the values of data items. { UPDATE REPLACE U REPLACE REPLACE REPL } ,data item name="value"; [data item name="value"; . . . . .] END
VERSION	Displays the current version of QUERY and IMAGE procedures and program files. { VERSION V }
XEQ	Executes QUERY commands from a file instead of the standard input device. { XEQ X } *filename* [,NODATA]

ASCII CHARACTER SET

APPENDIX B

GRAPHIC	COMMENTS	GRAPHIC	COMMENTS
	Null	*	Asterisk
	Start of heading	+	Plus
	Start of text	,	Comma
	End of text	-	Hyphen (minus)
	End of transmission	.	Period (Decimal)
	Enquiry	/	Slant
	Acknowledge	0	Zero
	Bell	1	One
	Backspace	2	Two
	Horizontal tabulation	3	Three
	Line feed	4	Four
	Vertical tabulation	5	Five
	Form feed	6	Six
	Carriage return	7	Seven
	Shift out	8	Eight
	Shift in	9	Nine
	Data link escape	:	Colon
	Device control 1	;	Semicolon
	Device control 2	<	Less than
	Device control 3	=	Equals
	Device control 4	>	Greater than
	Negative acknowledge	?	Question mark
	Synchronous idle	@	Commercial at
	End of transmission block	A	Uppercase A
	Cancel	B	Uppercase B
	End of medium	C	Uppercase C
	Substitute	D	Uppercase D
	Escape	E	Uppercase E
	File separator	F	Uppercase F
	Group separator	G	Uppercase G
	Record separator	H	Uppercase H
	Unit separator	I	Uppercase I
	Space	J	Uppercase J
!	Exclamation point	K	Uppercase K
"	Quotation mark	L	Uppercase L
#	Number sign	M	Uppercase M
$	Dollar sign	N	Uppercase N
%	Percent sign	O	Uppercase O
&	Ampersand	P	Uppercase P
'	Apostrophe	Q	Uppercase Q
(	Opening parenthesis	R	Uppercase R
)	Closing parenthesis	S	Uppercase S

GRAPHIC	COMMENTS	GRAPHIC	COMMENTS
T	Uppercase T	j	Lowercase j
U	Uppercase U	k	Lowercase k
V	Uppercase V	l	Lowercase l
W	Uppercase W	m	Lowercase m
X	Uppercase X	n	Lowercase n
Y	Uppercase Y	o	Lowercase o
Z	Uppercase Z	p	Lowercase p
[	Opening bracket	q	Lowercase q
\	Reverse slant	r	Lowercase r
]	Closing bracket	s	Lowercase s
^	Circumflex	t	Lowercase t
_	Underscore	u	Lowercase u
`	Grave accent	v	Lowercase v
a	Lowercase a	w	Lowercase w
b	Lowercase b	x	Lowercase x
c	Lowercase c	y	Lowercase y
d	Lowercase d	z	Lowercase z
e	Lowercase e	{	Opening (left) brace
f	Lowercase f	|	Vertical line
g	Lowercase g	}	Closing (right) brace
h	Lowercase h	~	Tilde
i	Lowercase i		Delete

EXCEPTIONAL ERROR CONDITIONS

APPENDIX C

The following error messages are printed only after an exceptional error condition occurs due to a hardware or software failure. If these conditions persist, they may require the aid of HP support personnel.

BAD DATA BASE NUMBER
BAD DATA SET OR DATA ITEM REFERENCE
BAD ITEM REFERENCE OR BAD LIST (SUB-SYSTEM ERROR)
BAD MODE
BEGINNING OF CHAIN
BEGINNING-OF-FILE
DATA BASE LOCKED BY ANOTHER PROCESS
DATA BASE NOT ENABLED
DATA ITEM LIST TOO LONG
DBML ERROR P1,P2,P3 (P1,P2,P3 are status returns from an IMAGE/3000 intrinsic.)
DIRECTED BEGINNING-OF-FILE
DIRECTED END-OF-FILE
DUPLICATE SEARCH ITEM VALUE
END OF CHAIN
END-OF-FILE
FCLOSE FAILURE
FOPEN FAILURE
FREADDIR FAILURE
FWRITEDIR FAILURE
FWRITELABEL FAILURE
FREADLABEL FAILURE
FLOCK FAILURE

ACCESSING A REMOTE DATA BASE

APPENDIX D

If you want to use QUERY to access a data base that resides on a remote HP 3000, there are three methods available. Method 1 is the most efficient since it minimizes communications line traffic.

METHOD 1: QUERY ON REMOTE SYSTEM

You can log on to your local computer, establish a communications link, initiate a remote session, and run QUERY on the remote system. This method is illustrated in figure D-1. A user whose user name is USERL and whose account is ACCTL logs on to the local system. He then establishes a communications link with the remote system referenced as dsdevice SYSR. A REMOTE HELLO command establishes a session on the remote system with user identification MEMBER.PAYROLL. The user runs QUERY on the remote system accessing a data base named DBR.

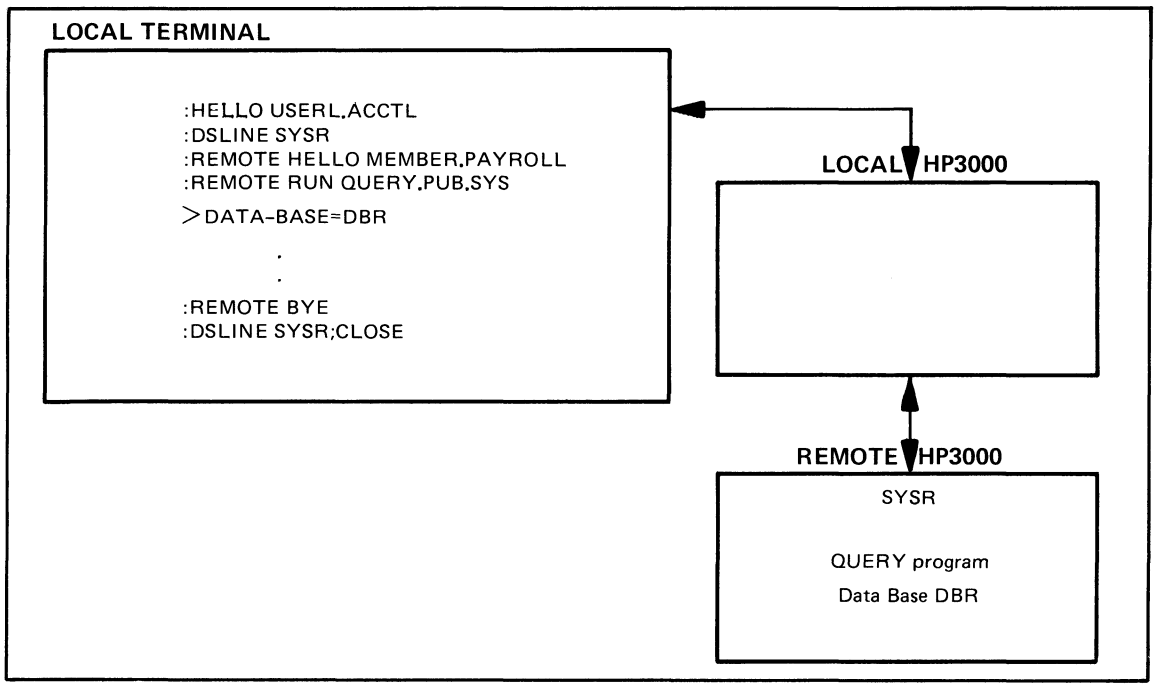


Figure D-1. Using Method 1

For complete instructions on operating remote sessions, refer to the *DS/3000 Reference Manual.*

METHOD 2: QUERY ON LOCAL SYSTEM

The second method is similar to Method 1 except QUERY is run on the local system. First a :FILE command is used to specify the location of the remote data base. After

initiating execution of QUERY, you specify the data base name with the DEFINE or DATA BASE= command. The name of the data base must be the same as the name specified as the formal file designator in the :FILE command.

Figure D-2 illustrates Method 2. After USERL establishes a communications link and remote session on the remote system, he enters a file equation to specify that the data base DBR resides on a remote system with dsdevice name SYSR. The device type may be omitted if the default device type is acceptable. The user then runs QUERY on the local system and specifies the data base name DBR with the DATA-BASE= command. Note that the data base name is the formal file designator in the :FILE command

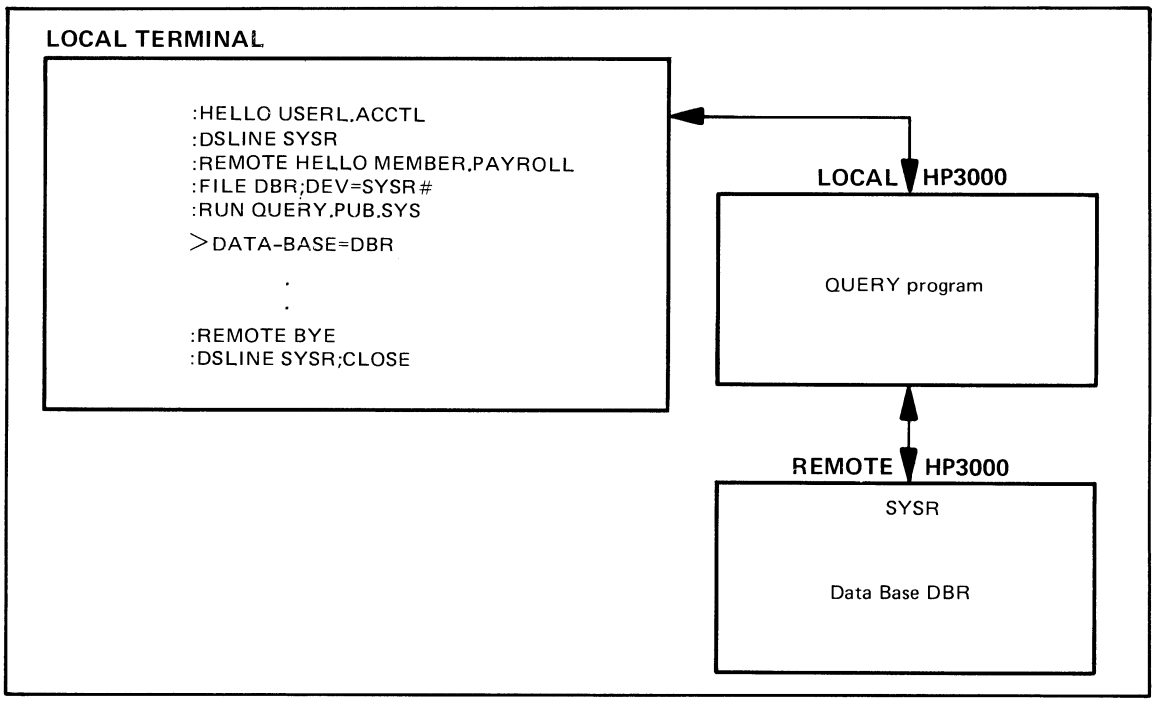


Figure D-2. Using Method 2

METHOD 3: USING A DATA-BASE-ACCESS FILE

The third method uses a data-base-access file to define the data base which is to be used. You must first log on to the local system and initiate QUERY execution. When specifying the data base name with the DEFINE or DATA BASE= command, you enter the name of a data-base-access file rather than a data base name. When IMAGE opens the data base, it recognizes the data-base-access file and programmatically establishes a communication link and remote session. If you use this method, QUERY automatically issues a REMOTE BYE and closes the DSLINE when you enter the QUERY EXIT command.

Data-base-access files are usually created by the data base administrator or manager. Detailed instructions for creating a data-base-access file are provided in the *IMAGE Reference Manual*, Section VIII.

Figure D-3 illustrates Method 3. The user logs on to the system and runs QUERY. He specifies a data-base-access file name with the DATA-BASE= command. The data-base-

access file provides the necessary information to enable QUERY to establish a communications link and remote session to access the data base DBR that resides on the remote system. The file contains the following records:

Record 1 FILE DBR;DEV=SYSR#DISCA
Record 2 DSLINE SYSR
Record 3 USERL.ACCTL=HELLO MEMBER.PAYROLL

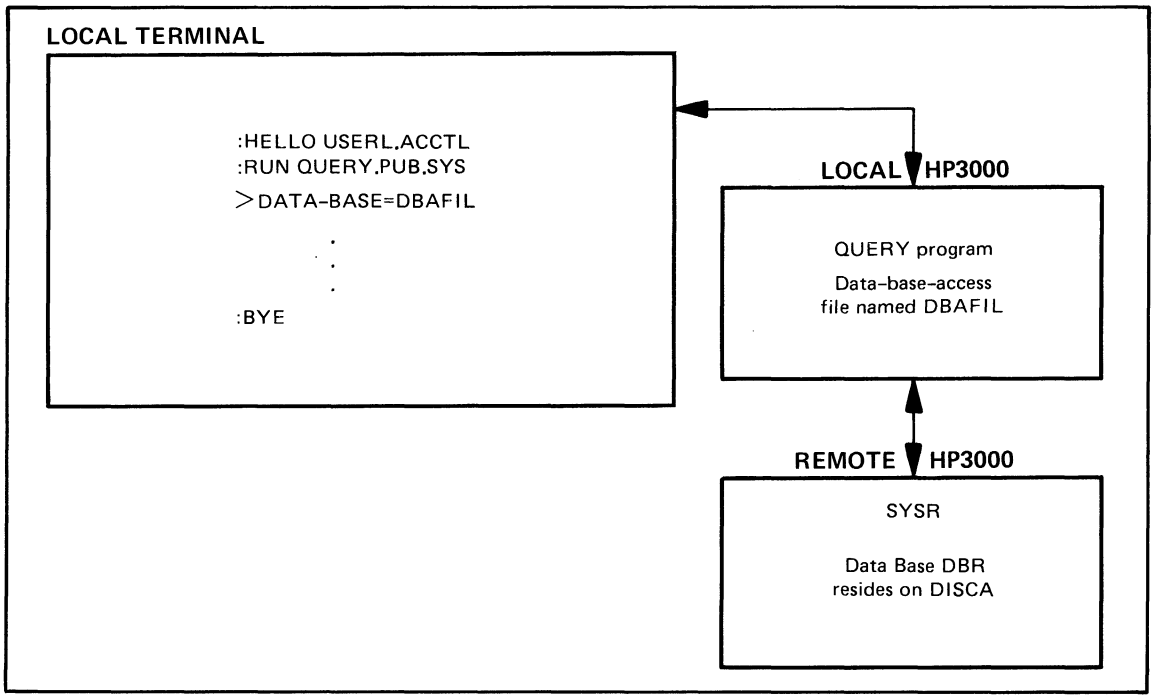


Figure D-3. Using Method 3

Record 1 of the data-base-access file specifies the remote system on which the data base resides and the device class of the disc (DISCA) on which it resides. Record 2 specifies the Distributed Systems (DS) line and Record 3 indicates the remote username and account name which is to be used to establish a remote session for USERL.ACCTL.

INDEX

A

access modes, 1-9, 2-8, 2-11
ADD command, see UPDATE ADD
ADD option, 4-32, 4-36
alphanumeric edit mask, 4-20
ALTER command, 5-10
ALTER command messages, 6-3
ampersand (&) continuation character, 1-9, 2-4, 5-3
AND connector, 3-4
ASC option, 4-25
ASCII characters, 1-12, B-1
ASCII files, 1-12, 5-2, 5-7
ASSIGN command, 2-5a
automatic data set list additions, 2-13
automatic master data sets, 1-6
AVERAGE option, 4-32

B

batch, 1-1
break key, 2-3, 2-19
:BYE command, 2-4

C

card reader, 2-4
chains, 3-12
character literal, 1-4
character set, QUERY, 1-13
commands, 1-1, 1-8, 1-9
  as procedures, 5-1
  summary, A-1
compound data items, 1-6
continuation character (&), 1-9, 2-4, 5-3
control break, 4-27, 4-30, 4-33
control H, 2-1
control X, 2-1
control Y, 2-3, 3-16, 5-10
correcting REPORT statements, 4-44
COUNT option, 4-32
CREATE command, 5-2
  SPACE, 5-6
CREATE command messages, 6-4

D

data base access, 1-1
data-base-access file, 2-8,D-1
data base administrator, 1-2
DATA-BASE= command, 2-1, 2-4, 2-8
DATA-BASE= command messages, 6-5
data base designer, 1-2, 1-4, 1-6
data base structure, 1-2, 1-7
data entry, 1-3
data item, 1-2
  fully-qualified names, 1-3
data set, 1-2
data set list, 2-2, 2-12
data set selection rules, 2-12, 4-6
DATA-SETS= command, 2-12
DATA-SETS= command messages, 6-7
data types, 1-4, 1-5
  and registers, 4-38
data values, 1-4
DATE option, 4-16
DEFINE command, 2-1, 2-6
DELETE command, see UPDATE DELETE command
delete statement (/D), 5-13
DES option, 4-25
designing a report, 4-14
DESTROY command, 5-15
DESTROY command messages, 6-7
detail data set, 1-6, 3-11, 3-16
detail search item, 3-11
detail statement, 4-18
device class (type), 1-12, 2-18
DISPLAY command, 5-7
  DISPLAY LIST, 5-9
DISPLAY command messages, 6-7
Distributed systems (DS/3000), 2-8,D-1
DIVIDE option, 4-36

E

edit mask, 4-20
edit statement, 4-20
EDIT/3000, 5-5
end statement (/E), 5-14
environment, access mode, 1-11
environment commands, 2-2
:EOJ command, 2-22
error messages, 6-1
exceptional error conditions, C-1
EXIT command, 2-22
extended precision, 1-5

F

field widths, LIST output, 4-4
:FILE command, 1-12, 2-18, 4-10
file, displaying to a, 5-7
file input, CREATE, 5-3

file, QUERY commands from a, 5-1
FIND command, 2-12, 3-1, 3-3
  FIND ALL, 3-9
  FIND CHAIN, 3-11
  FIND procedure, 3-10
  relational operators, 3-4
FIND command messages, 6-8
foldout, 4-45
FORM command, 1-3, 2-23
FORM command messages, 6-11
format, LIST output, 4-4
FORMAT option, 2-28
fully-qualified names, 1-3
FUNCTION option, 2-28

G

general messages, 6-1
group statement, 4-29

H

H^C, 2-1
header statement, 4-16
:HELLO command, 2-1
HELP command, 2-3, 2-28

I

IMAGE/3000, 1-1, 1-6, 6-1
initializing registers, 4-37
insert statement (/I), 5-11
interactive mode, 2-1
ITEMS option, 2-23

J

job, 1-1, 2-1, 2-4, 3-21
:JOB command, 2-4

K

key item, 1-6, 2-24
key words, 1-1

L

line printer, 2-4
LINES= statement, 4-42
LIST command, 2-12, 4-1, 4-2
LIST command messages, 6-11
literal, 1-4
LOAD option, 4-36
locking data base, 1-10
LOCKOPTION, 2-5a, 2-29a
logging, 1-12
logical connectors, 3-4, 4-4
log-on procedure, 2-1
LP device class, 1-12
LP option, 2-18

M

major to minor sort fields, 4-27
manual master data set, 1-6, 3-16
master data sets, 3-11, 3-18
master search item, 3-11
missing chain head, 3-16
MODE= command, 2-11
modes of access, 1-9
MPE, 2-1
  files, 1-12
multiple search values, 3-5
MULTIPLY option, 4-36

N

negative number representation, 4-10
NODATA option, 5-17
NOPAGE statement, 4-42
null values, 3-6, 3-15, 3-21
numeric edit masks, 4-21
  characters, 4-22
  combinations, 4-23
numeric literal, 1-4, 4-39

O

OR connector, 3-4
OUT statement, 4-42
OUTPUT= command, 2-18, 2-24, 2-29
OUTPUT= command messages, 6-13
output control statements, 4-42
output file, 2-2

P

packed decimal number, 1-4,1-5
PAGENO option, 4-16
PARAMETERS option, 2-28
passwords, 1-2, 2-8
PASSWORD= command, 2-10
PATHS option, 2-23
PAUSE statement, 4-42
procedure, 1-1, 2-16, 5-1
Proc-file, 1-1, 2-2, 2-16, 5-1
PROC-FILE= command, 2-4, 2-16, 5-1
PROC-FILE= command messages, 6-14

Q

QSLIST file, 1-12, 2-18, 2-24, 2-29, 2-30, 5-9
QUERY registers, using, 4-37

R

real number values, 1-5, 3-15a, 4-38
  report output, 4-21
register statement, 4-36
  initializing, 4-37

relational operators, 3-4, 4-3
remote data base access, 2-8,D-1
RENAME command, 5-16
RENAME command messages, 6-15
REPLACE command, see UPDATE REPLACE
replace statement (/R), 5-12
REPORT command, 4-1, 4-13
  REPORT ALL, 4-10
  REPORT procedure, 4-44
REPORT command messages, 6-15
:RESUME command, 2-3
:RUN command, 2-1

S

sample data base, 1-6
search item, 1-6, 2-24, 3-11
security, 1-1, 3-18
serial searching, 3-2
session, 1-1, 2-1, 3-21
SETS option, 2-23
SHOW command, 2-29a
SKIP option, 4-18, 4-32, 4-45, 4-45b
slash character, 3-16, 5-3
sort item, 1-6, 2-24
sort statement, 4-25
SPACE option, 4-16, 4-18, 4-32, 4-45, 4-45b
special characters, 1-12
standard list device, 2-29, 2-30
statement parameters summary, 4-45
$STDLIST, 1-12, 2-4, 2-18, 4-39, 4-42, 6-1
STORE data base, 1-6
structure of data base, 1-2, 1-7
sub-items, 1-6
SUBTRACT option, 4-36
summary of QUERY commands, A-1
SYSDUMP, 1-10

T

TAPE option, 2-19
TERM option, 2-18
terminal input, CREATE, 5-3
terminating commands, 3-16, 5-3
TIME option, 4-16
total statement, 4-32

U

U type values, 3-4, 3-15
unlocking data base, 1-10
UPDATE ADD command, 3-1, 3-14
UPDATE command messages, 6-21
UPDATE DELETE command, 3-1, 3-18
UPDATE procedure, 3-24
UPDATE REPLACE command, 3-1, 3-20
using serial read, 3-2

V

VERSION command, 2-30

X

X^C, 2-1
X type values, 3-4, 3-15
XEQ command, 5-1, 5-17
XEQ command messages, 6-25

Y

Y^C, 2-3, 3-16, 5-10

Z

zoned decimal number, 1-5

READER COMMENT SHEET

HP 3000 Series II Computer System
QUERY
Reference Manual

30000-90042 June 1976

We welcome your evaluation of this manual. Your comments and suggestions help us improve our publications. Please use additional pages if necessary.

Is this manual technically accurate?

Did you have any difficulty in understanding concepts or wording? Where?

Is the format of this manual convenient in size, arrangement, and readability? What improvements would you suggest?

Other comments?

FROM:

Name _____

Company _____

Address _____

_____

_____

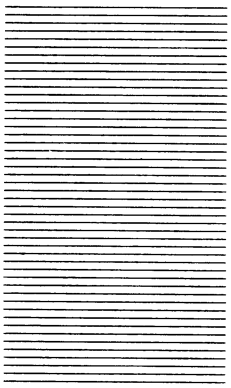
FOLD FOLD

FIRST CLASS
PERMIT NO. 1020
SANTA CLARA
CALIFORNIA

BUSINESS REPLY MAIL

No Postage Necessary if Mailed in the United States. Postage will be paid by

Publications Manager, Product Support Group
Hewlett-Packard Company
General Systems Division
5303 Stevens Creek Boulevard
Santa Clara, California 95050



FOLD FOLD

SALES OFFICES
Arranged alphabetically by country

1

ANGOLA
Telectra
Empresa Técnica de Equipamentos Eléctricos, S.A.R.L.
R. Barbosa Rodrigues, 41-1°DT.°
Caixa Postal, 6487
Luanda
Tel: 35515/6

ARGENTINA
Hewlett-Packard Argentina S.A.
Santa Fe 2035, Martinez
6140 **Buenos Aires**
Tel: 792-1239, 798-6086
Telex: 122443 AR CIGY

Biotron S.A.C.I.y M.
Avda. Paseo Colon 221
9 piso
1399 **Buenos Aires**
Tel: 30-4846/1851/8384
34-9356/0460/4551
Telex: (33) 17595 BIO AR

AUSTRALIA

AUSTRALIA CAPITAL TERR.
Hewlett-Packard Australia Pty. Ltd.
121 Wollongong Street
Fyshwick, 2609
Tel: 804244
Telex: 62650

NEW SOUTH WALES
Hewlett-Packard Australia Pty. Ltd.
31 Bridge Street
Pymble, 2073
Tel: 4496566
Telex: 21561

QUEENSLAND
Hewlett Packard Australia Pty. Ltd.
5th Floor
Teachers Union Building
495-499 Boundary Street
Spring Hill, 4000
Tel: 2291544

SOUTH AUSTRALIA
Hewlett-Packard Australia Pty. Ltd.
153 Greenhill Road
Parkside, 5063
Tel: 2725911
Telex: 82536

VICTORIA
Hewlett-Packard Australia Pty. Ltd.
31-41 Joseph Street
Blackburn, 3130
Tel: 89-6351
Telex: 31024 MELB

WESTERN AUSTRALIA
Hewlett-Packard Australia Pty. Ltd.
141 Stirling Highway
Nedlands, 6009
Tel: 3865455
Telex: 93859

AUSTRIA
Hewlett-Packard Ges.m.b.H.
Wehlistrasse 29
P.O. Box 7
A-1205 **Vienna**
Tel: 35-16-21-0
Telex: 13582/135066

Hewlett-Packard Ges.m.b.H.
Wehlistrasse, 29
A-1205 **Wien**
Tel: 35-16-21
Telex: 135066

BAHRAIN
Medical Only
Wael Pharmacy
P.O. Box 648
Bahrain
Tel: 54886, 56123
Telex: 8550 WAEL GJ

Al Hamidiya Trading and Contracting
P.O. Box 20074
Manama
Tel: 259978, 259958
Telex: 8895 KALDIA GJ

BANGLADESH
The General Electric Co. of Bangladesh Ltd.
Magnet House 72
Dilkusha Commercial Area
Motijheil, Dacca 2
Tel: 252415, 252419
Telex: 734

BELGIUM
Hewlett-Packard Benelux S.A./N.V.
Avenue du Col-Vert, 1, (Groenkraaglaan)
B-1170 **Brussels**
Tel: (02) 660 50 50
Telex: 23-494 paloben bru

BRAZIL
Hewlett-Packard do Brasil I.e.C. Ltda.
Alameda Rio Negro, 750
Alphaville
06400 **Barueri** SP
Tel: 429-3222

Hewlett-Packard do Brasil I.e.C. Ltda.
Rua Padre Chagas, 32
90000-**Pôrto Alegre**-RS
Tel: 22-2998, 22-5621

Hewlett-Packard do Brasil I.e.C. Ltda.
Av. Epitacio Pessoa, 4664
22471-**Rio de Janeiro**-RJ
Tel: 286-0237
Telex: 021-21905 HPBR-BR

CANADA

ALBERTA
Hewlett-Packard (Canada) Ltd.
11620A - 168th Street
Edmonton T5M 3T9
Tel: (403) 452-3670
TWX: 610-831-2431

Hewlett-Packard (Canada) Ltd.
210, 7220 Fisher St. S.E.
Calgary T2H 2H8
Tel: (403) 253-2713
TWX: 610-821-6141

BRITISH COLUMBIA
Hewlett-Packard (Canada) Ltd.
10691 Shellbridge Way
Richmond V6X 2W7
Tel: (604) 270-2277
TWX: 610-925-5059

MANITOBA
Hewlett-Packard (Canada) Ltd.
380-550 Century St.
St. James,
Winnipeg R3H 0Y1
Tel: (204) 786-6701
TWX: 610-671-3531

NOVA SCOTIA
Hewlett-Packard (Canada) Ltd.
P.O. Box 931
800 Windmill Road
Dartmouth B3B 1L1
Tel: (902) 469-7820
TWX: 610-271-4482

ONTARIO
Hewlett-Packard (Canada) Ltd.
1020 Morrison Dr.
Ottawa K2H 8K7
Tel: (613) 820-6483
TWX: 610-563-1636

Hewlett-Packard (Canada) Ltd.
6877 Goreway Drive
Mississauga L4V 1M8
Tel: (416) 678-9430
TWX: 610-492-4246

Hewlett-Packard (Canada) Ltd.
552 Newbold Street
London N6E 2S5
Tel: (519) 686-9181
TWX: 610-352-1201

QUEBEC
Hewlett-Packard (Canada) Ltd.
275 Hymus Blvd.
Pointe Claire H9R 1G7
Tel: (514) 697-4232
TWX: 610-422-3022

FOR CANADIAN AREAS NOT LISTED:
Contact Hewlett-Packard (Canada) Ltd. in Mississauga.

CHILE
Jorge Calcagni y Cia. Ltda.
Arturo Burhle 065
Casilla 16475
Correo 9, **Santiago**
Tel: 220222
Telex: JCALCAGNI

COLOMBIA
Instrumentación
Henrik A. Langebaek & Kier S.A.
Carrera 7 No. 48-75
Apartado Aéreo 6287
Bogotá, 1 D.E.
Tel: 269-8877
Telex: 44400

Instrumentación
H.A. Langebaek & Kier S.A.
Carrera 63 No. 49-A-31
Apartado 54098
Medellin
Tel: 304475

COSTA RICA
Cientifica Costarricense S.A.
Avenida 2, Calle 5
San Pedro de Montes de Oca
Apartado 10159
San José
Tel: 24-38-20, 24-08-19
Telex: 2367 GALGUR CR

CYPRUS
Kypronics
19 Gregorios Xenopoulos Street
P.O. Box 1152
Nicosia
Tel: 45628/29
Telex: 3018

CZECHOSLOVAKIA
Hewlett-Packard
Obchodni zastupitelstvi v CSSR
Pisemny styk
Post. schranka 27
CS 118 01 **Praha** 011
CSSR

Vyvojova a Provozni Zakladna
Vyzkumnych Ustavu v Bechovicich
CSSR-25097 **Bechovice u Prahy**
Tel: 89 93 41
Telex: 12133

Institute of Medical Bionics
Vyskumny Ustav Lekarskej Bioniky
Jedlova 6
CS-88346 **Bratislava-Kramare**
Tel: 44-551
Telex: 93229

DENMARK
Hewlett-Packard A/S
Datavej 52
DK-3460 **Birkerod**
Tel: (02) 81 66 40
Telex: 37409 hpas dk

Hewlett-Packard A/S
Navervej 1
DK-8600 **Silkeborg**
Tel: (06) 82 71 66
Telex: 37409 hpas dk

ECUADOR
CYEDE Cia. Ltda.
P.O. Box 6423 CCI
Av. Eloy Alfaro 1749
Quito
Tel: 450-975, 243-052
Telex: 2548 CYEDE ED

Medical Only
Hospitalar S.A.
Casilla 3590
Robles 625
Quito
Tel: 545-250

EGYPT
I.E.A.
International Engineering Associates
24 Hussein Hegazi Street
Kasr-el-Aini
Cairo
Tel: 23 829
Telex: 93830

SAMITRO
Sami Amin Trading Office
18 Abdel Aziz Gawish
Abdine-Cairo
Tel: 24932

EL SALVADOR
IPESA
Bulevar de los Heroes 11-48
Edificio Sarah 1148
San Salvador
Tel: 252787

ETHIOPIA
Abdella Abdulmalik
P.O. Box 2635
Addis Ababa
Tel: 11 93 40

FINLAND
Hewlett-Packard Oy
Revontulentie, 7
SF-02100 **Espoo** 10
Tel: (90) 455 0211
Telex: 121563 hewpa sf

FRANCE
Hewlett-Packard France
Zone d'activites de Courtaboeuf
Avenue des Tropiques
Boite Postale 6
91401 **Orsay**-Cédex
Tel: (1) 907 78 25
TWX: 600048F

Hewlett-Packard France
Chemin des Mouilles
B.P. 162
69130 **Ecully**
Tel: (78) 33 81 25
TWX: 310617F

Hewlett-Packard France
20, Chemin de La Cépière
31081 **Toulouse**
Le Mirail-Cédex
Tel: (61) 40 11 12

Hewlett-Packard France
Le Ligoures
Place Romée de Villeneuve
13100 **Aix-en-Provence**
Tel: (42) 59 41 02
TWX: 410770F

Hewlett-Packard France
2, Allee de la Bourgonette
35100 **Rennes**
Tel: (99) 51 42 44
TWX: 740912F

Hewlett-Packard France
18, rue du Canal de la Marne
67300 **Schiltigheim**
Tel: (88) 83 08 10
TWX: 890141F

Hewlett-Packard France
Immeuble péricentre
rue van Gogh
59650 **Villeneuve** D'Ascq
Tel: (20) 91 41 25
TWX: 160124F

Hewlett-Packard France
Bâtiment Ampère
Rue de la Commune de Paris
B.P. 300
93153 **Le Blanc Mesnil**-Cédex
Tel: (01) 931 88 50
Telex: 211032F

Hewlett-Packard France
Av. du Pdt. Kennedy
33700 **Merignac**
Tel: (56) 97 01 81

Hewlett-Packard France
Immeuble Lorraine
Boulevard de France
91035 **Evry**-Cédex
Tel: 077 96 60
Telex: 692315F

Hewlett-Packard France
23 Rue Lothaire
57000 **Metz**
Tel: (87) 65 53 50

GERMAN FEDERAL REPUBLIC
Hewlett-Packard GmbH
Vertriebszentrale Frankfurt
Berner Strasse 117
Postfach 560 140
D-6000 **Frankfurt** 56
Tel: (06011) 50041
Telex: 04 13249 hpffm d

Hewlett-Packard GmbH
Technisches Büro Böblingen
Herrenberger Strasse 110
D-7030 **Böblingen,**
Württemberg
Tel: (07031) 667-1
Telex: 07265739 bbn

Hewlett-Packard GmbH
Technisches Büro Düsseldorf
Emanuel-Leutze-Str. 1 (Seestern)
D-4000 **Düsseldorf**
Tel: (0211) 5971-1
Telex: 085/86 533 hpdd d

Hewlett-Packard GmbH
Technisches Büro Hamburg
Kapstadtring 5
D-2000 **Hamburg** 60
Tel: (040) 63804-1
Telex: 21 63 032 hphh d

Hewlett-Packard GmbH
Technisches Büro Hannover
Am Grossmarkt 6
D-3000 **Hannover** 91
Tel: (0511) 46 60 01
Telex: 092 3259

Hewlett-Packard GmbH
Technisches Büro Nürnberg
Neumeyerstrasse 90
D-8500 **Nürnberg**
Tel: (0911) 52 20 83
Telex: 0623 860

Hewlett-Packard GmbH
Technisches Büro München
Eschenstrasse 5
D-8021 **Taufkirchen**
Tel: (089) 6117-1
Telex: 0524985

Hewlett-Packard GmbH
Technisches Büro Berlin
Kaithstrasse 2-4
D-1000 **Berlin** 30
Tel: (030) 24 90 86
Telex: 018 3405 hpbln d

GREECE
Kostas Karayannis
8 Omirou Street
Athens 133
Tel: 32 30 303/32/37 731
Telex: 21 59 62 RKAR GR

GUAM
Guam Medical Supply, Inc.
Suite C, Airport Plaza
P.O. Box 8947
Tamuning 96911
Tel: 646-4513

GUATEMALA
IPESA
Avenida Reforma 3-48
Zona 9
Guatemala City
Tel: 316627, 314786, 66471-5, ext. 9
Telex: 4192 Teletro Gu

HONG KONG
Hewlett-Packard Hong Kong Ltd.
11th Floor, Four Seas Bldg.
212 Nathan Rd.
Kowloon
Tel: 3-697446 (5 lines)
Telex: 36678 HX

Medical/Analytical Only
Schmidt & Co. (Hong Kong) Ltd.
Wing On Centre, 28th Floor
Connaught Road, C.
Hong Kong
Tel: 5-455644
Telex: 74766 SCHMX HX

INDIA
Blue Star Ltd.
Sahas
414/2 Vir Savarkar Marg
Prabhadevi
Bombay 400 025
Tel: 45 78 87
Telex: 011-4093

Blue Star Ltd.
Band Box House
Prabhadevi
Bombay 400 025
Tel: 45 73 01
Telex: 011-3751

Blue Star Ltd.
Bhavdeep
Stadium Road
Ahmedabad 380 014
Tel: 43922
Telex: 012-234

Blue Star Ltd.
7 Hare Street
Calcutta 700 001
Tel: 23-0131
Telex: 021-7655

Blue Star Ltd.
Bhandari House
91 Nehru Place
New Delhi 110 024
Tel: 682547
Telex: 031-2463

Blue Star Ltd.
T.C. 7/603 'Poornima'
Maruthankuzhi
Trivandrum 695 013
Tel: 65799
Telex: 0884-259

Blue Star Ltd.
11 Magarath Road
Bangalore 560 025
Tel: 55668
Telex: 0845-430

Blue Star Ltd.
Meeakshi Mandiram
XXXXV/1379-2 Mahatma Gandhi Rd.
Cochin 682 016
Tel: 32069
Telex: 085-514

Blue Star Ltd.
1-1-117/1 Sarojini Devi Road
Secunderabad 500 033
Tel: 70126
Telex: 0155-459

Blue Star Ltd.
133 Kodambakkam High Road
Madras 600 034
Tel: 82057
Telex: 041-379

ICELAND
Medical Only
Elding Trading Company Inc.
Hafnarnvoli - Tryggvagötu
P.O. Box 895
IS-**Reykjavik**
Tel: 1 58 20/1 63 03

INDONESIA
BERCA Indonesia P.T.
P.O. Box 496/Jkt.
Jln. Abdul Muis 62
Jakarta
Tel: 349255, 349886
Telex: 46748 BERSIL IA

BERCA Indonesia P.T.
P.O. Box 174/Sby.
23 Jln. Jimerto
Surabaya
Tel: 42027

IRELAND
Hewlett-Packard Ltd.
Kestrel House
Clanwilliam Place
Lower Mount Street
Dublin 2, Eire

Hewlett-Packard Ltd.
2C Avongberg Ind. Est.
Long Mile Road
Dublin 12
Tel: 514322/514224
Telex: 30439

Medical Only
Cardiac Services (Ireland) Ltd.
Kilmore Road
Artane
Dublin 5, Eire
Tel: (01) 315820

Medical Only
Cardiac Services Co.
95A Finaghy Rd. South
Belfast BT10 0BY
GB-Northern Ireland
Tel: (0232) 625566
Telex: 747626

ISRAEL
Electronics Engineering Div. of Motorola Israel Ltd.
16, Kremenetski Street
P.O. Box 25016
Tel-Aviv
Tel: 38973
Telex: 33569, 34164

ITALY
Hewlett-Packard Italiana S.p.A.
Via G. Di Vittorio, 9
20063 **Cernusco Sul Naviglio** (MI)
Tel: (2) 903691
Telex: 334632 HEWPACKIT

Hewlett-Packard Italiana S.p.A.
Via Turazza, 14
35100 **Padova**
Tel: (49) 664888
Telex: 430315 HEWPACKI

Hewlett-Packard Italiana S.p.A.
Via G. Armellini 10
1-00143 **Roma**
Tel: (06) 54 69 61
Telex: 610514

Hewlett-Packard Italiana S.p.A.
Corso Giovanni Lanza 94
I-10133 **Torino**
Tel: (011) 659308
Telex: 221079

Hewlett-Packard Italiana S.p.A.
Via Principe Nicola 43 G/C
I-95126 **Catania**
Tel: (095) 37 05 04
Telex: 970291

Hewlett-Packard Italiana S.p.A.
Via Nuova san Rocco A Capadimonte, 62A
80131 **Napoli**
Tel: (081) 710698

Hewlett-Packard Italiana S.p.A.
Via Martin Luther King, 38/111
I-40132 **Bologna**
Tel: (051) 402394
Telex: 511630

JAPAN
Yokogawa-Hewlett-Packard Ltd.
29-21, Takaido-Higashi 3-chome
Suginami-ku, **Tokyo** 168
Tel: 03-331-6111
Telex: 232-2024 YHP-Tokyo

Yokogawa-Hewlett-Packard Ltd.
Chuo Bldg., 4th Floor
4-20, Nishinakajima 5-chome
Yodogawa-ku, Osaka-shi
Osaka, 532
Tel: 06-304-6021
Telex: 523-3624

Yokogawa-Hewlett-Packard Ltd.
Sunitomo Seimei Nagaya Bldg.
11-2 Shimosasajima-cho,
Nakamura-ku, **Nagoya,** 450
Tel: 052 571-5171

Yokogawa-Hewlett-Packard Ltd.
Tanigawa Building
2-24-1 Tsuruya-cho
Kanagawa-ku
Yokohama, 221
Tel: 045-312-1252
Telex: 382-3204 YHP YOK

Yokogawa-Hewlett-Packard Ltd.
Mito Mitsui Building
105, 1-chome, San-no-maru
Mito, Ibaragi 310
Tel: 0292-25-7470

Yokogawa-Hewlett-Packard Ltd.
Inoue Building
1348-3, Asahi-cho, 1-chome
Atsugi, Kanagawa 243
Tel: 0462-24-0452

Yokogawa-Hewlett-Packard Ltd.
Kumagaya Asahi
Hachijuni Building
4th Floor
3-4, Tsukuba
Kumagaya, Saitama 360
Tel: 0485-24-6563

JORDAN
Mouasher Cousins Co.
P.O. Box 1387
Amman
Tel: 24907/39907
Telex: SABCO JO 1456

KENYA
ADCOM Ltd., Inc.
P.O. Box 30070
Nairobi
Tel: 331955
Telex: 22639

Medical Only
International Aeradio (E.A.) Ltd.
P.O. Box 19012
Nairobi Airport
Nairobi
Tel: 336055/56
Telex: 22201/22301

Medical Only
International Aeradio (E.A.) Ltd.
P.O. Box 95221
Mombasa

KOREA
Samsung Electronics Co., Ltd.
4759 Shingil-6-Dong
Yeong Deung POU
Seoul
Tel: 833-4122, 4121
Telex: SAMSAN 27364

KUWAIT
Al-Khaldiya Trading & Contracting
P.O. Box 830-Safat
Kuwait
Tel: 42 4910/41 1726
Telex: 2481 Areeg kt

LUXEMBURG
Hewlett-Packard Benelux S.A./N.V.
Avenue du Col-Vert, 1 (Groenkraaglaan)
B-1170 **Brussels**
Tel: (02) 660 5050
Telex: 23 494

MALAYSIA
Hewlett-Packard Sales (Malaysia) Sdn. Bhd.
Suite 2.21/2.22
Bangunan Angkasa Raya
Jalan Ampang
Kuala Lumpur
Tel: 483680, 485653

Protel Engineering
P.O. Box 1917
Lot 259, Satok Road
Kuching, **Sarawak**
Tel: 53544

MEXICO
Hewlett-Packard Mexicana, S.A. de C.V.
Av. Periférico Sur No. 6501
Tepepan, Xochimilco
Mexico 23, D.F.
Tel: 905-676-4600
Telex: 017-74-507

Hewlett-Packard Mexicana, S.A. de C.V.
Rio Volga #600
Col. Del Valle
Monterrey, N.L.
Tel: 78-32-10

MOROCCO
Dolbeau
81 rue Karatchi
Casablanca
Tel: 3041 82
Telex: 23051/22822

Gerep
2, rue d'Agadir
Boite Postal 156
Casablanca
Tel: 272093/5
Telex: 23 739

MOZAMBIQUE
A.N. Goncalves, Ltd.
162, 1° Apt. 14 Av. D. Luis
Caixa Postal 107
Maputo
Tel: 27091, 27114
Telex: 6-203 NEGON Mo

NETHERLANDS
Hewlett-Packard Benelux N.V.
Van Heuven Goedhartlaan 121
P.O. Box 667
1181KK **Amstelveen**
Tel: (20) 47 20 21
Telex: 13 216

NEW ZEALAND
Hewlett-Packard (N.Z.) Ltd.
4-12 Cruickshank Street
Kilbirnie, Wellington 3
P.O. Box 9443
Courtney Place
Wellington
Tel: 877-199

Hewlett-Packard (N.Z.) Ltd.
P.O. Box 26-189
169 Manukau Road
Epsom, **Auckland**
Tel: 687-159

Analytical/Medical Only
Northrop Instruments & Systems Ltd.,
Sturdee House
85-87 Ghuznee Street
P.O. Box 2406
Wellington
Tel: 850-091
Telex: NZ 31291

Northrup Instruments & Systems Ltd.
Eden House, 44 Khyber Pass Rd.
P.O. Box 9682, Newmarket
Auckland 1
Tel: 794-091

Northrup Instruments & Systems Ltd.
Terrace House, 4 Oxford Terrace
P.O. Box 8388
Christchurch
Tel: 64-165

NIGERIA
The Electronics Instrumentations Ltd.
N6B/770 Oyo Road
Okuseun House
P.M.B. 5402
Ibadan
Tel: 461577
Telex: 31231 TEIL NG

The Electronics Instrumentations Ltd.
144 Agege Motor Road, Mushin
P.O. Box 481
Mushin, **Lagos**

NORWAY
Hewlett-Packard Norge A/S
Ostendalen 18
P.O. Box 34
1345 **Osteraas**
Tel: (02) 1711 80
Telex: 16621 hpnas n

Hewlett-Packard Norge A/S
Nygaardsgaten 114
P.O. Box 4210
5013 Nygaardsgaten,
Bergen
Tel: (05) 21 97 33

PANAMA
Electrónico Balboa, S.A.
Aparatado 4929
Panama 5
Calle Samuel Lewis
Edificio "Alfa," No. 2
Ciudad de Panama
Tel: 64-2700
Telex: 3483103 Curundu, Canal Zone

PERU
Compañía Electro Médica S.A.
Los Flamencos 145
San Isidro Casilla 1030
Lima 1
Tel: 41-4325
Telex: Pub. Booth 25424 SISIDRO

PAKISTAN
Mushko & Company Ltd.
Oosman Chambers
Abdullah Haroon Road
Karachi-3
Tel: 511027, 512927
Telex: 2894

SALES OFFICES
Arranged alphabetically by country (cont.)

Mushko & Company, Ltd.
10, Bazar Rd.
Sector G-6/4
Islamabad
Tel: 28264

PHILIPPINES
The Online Advanced Systems Corporation
Rico House
Amorsolo cor. Herrera Str.
Legaspi Village, Makati
P.O. Box 1510
Metro **Manila**
Tel: 85-35-81, 85-34-91, 85-32-21
Telex: 3274 ONLINE

RHODESIA
Field Technical Sales
45 Kelvin Road North
P.O. Box 3458
Salisbury
Tel: 705231 (5 lines)
Telex: RH 4122

POLAND
Biuro Informacji Technicznej
Hewlett-Packard
Ul Stawki 2, 6P
PL00-950 **Warszawa**
Tel: 39 59 62, 39 51 87
Telex: 81 24 53

PORTUGAL
Telectra-Empresa Técnica de Equipamentos Eléctricos S.a.r.l.
Rua Rodrigo da Fonseca 103
P.O. Box 2531
P-**Lisbon** 1
Tel: (19) 68 60 72
Telex: 12598

Medical Only
Mundinter
Intercambio Mundial de Comércio S.a.r.l.
P.O. Box 2761
Avenida Antonio Augusto de Aguiar 138
P-**Lisbon**
Tel: (19) 53 21 31/7
Telex: 16691 munter p

PUERTO RICO
Hewlett-Packard Inter-Americas
Puerto Rico Branch Office
Calle 272,
#203 Urb. Country Club
Carolina 00630
Tel: (809) 762-7255
Telex: 345 0514

QATAR
Nasser Trading & Contracting
P.O. Box 1563
Doha
Tel: 22170
Telex: 4439 NASSER

ROMANIA
Hewlett-Packard
Reprezentanta
Bd.n. Balcescu 16
Bucuresti
Tel: 15 80 23/13 88 85
Telex: 10440

SAUDI ARABIA
Modern Electronic Establishment (Head Office)
P.O. Box 1228, Baghdadiah Street
Jeddah
Tel: 27 798
Telex: 40035
Cable: ELECTA JEDDAH

Modern Electronic Establishment (Branch)
P.O. Box 2728
Riyadh
Tel: 62596/66232
Telex: 202049

Modern Electronic Establishment (Branch)
P.O. Box 193
Al-Khobar
Tel: 44678-44813
Telex: 670136
Cable: ELECTA AL-KHOBAR

SINGAPORE
Hewlett-Packard Singapore (Pte.) Ltd.
6th Floor, Inchcape House
450-452 Alexandra Road
P.O. Box 58
Alexandra Post Office
Singapore 9115
Tel: 631788
Telex: HPSG RS 21486

SOUTH AFRICA
Hewlett-Packard South Africa (Pty.), Ltd.
Private Bag Wendywood,
Sandton, Transvaal, 2144
Hewlett-Packard Centre
Daphne Street, Wendywood,
Sandton, 2144
Tel: 802-5111/25
Telex: 8-4782

Hewlett-Packard South Africa (Pty.), Ltd.
P.O. Box 120
Howard Place,
Cape Province, 7450
Pine Park Centre, Forest Drive,
Pinelands,
Cape Province, 7405
Tel: 53-7955 thru 9
Telex: 57-0006

SPAIN
Hewlett-Packard Española, S.A.
Calle Jerez 3
E-**Madrid** 16
Tel: (1) 458 26 00 (10 lines)
Telex: 23515 hpe

Hewlett-Packard Española S.A.
Colonia Mirasierra
Edificio Juban
c/o Costa Brava, 13
Madrid 34

Hewlett-Packard Española, S.A.
Milanesado 21-23
E-**Barcelona** 17
Tel: (3) 203 6200 (5 lines)
Telex: 52603 hpbe e

Hewlett-Packard Española, S.A.
Av Ramón y Cajal, 1
Edificio Sevilla, planta 9°
E-**Sevilla** 5
Tel: 64 44 54/58

Hewlett-Packard Española S.A.
Edificio Albia II 7° B
E-**Bilbao** 1
Tel: 23 83 06/23 82 06

Hewlett-Packard Española S.A.
C/Ramon Gordillo 1
(Entlo.)
E-**Valencia** 10
Tel: 96-361.13.54/361.13.58

SRI LANKA
Metropolitan Agencies Ltd.
209/9 Union Place
Colombo 2
Tel: 35947
Telex: 1377METROLTD CE

SUDAN
Radison Trade
P.O. Box 921
Khartoum
Tel: 44048
Telex: 375

SURINAM
Surtel Radio Holland N.V.
Grote Hofstr. 3-5
P.O. Box 155
Paramaribo
Tel: 72118, 77880

SWEDEN
Hewlett-Packard Sverige AB
Enighetsvägen 3, Fack
S-161 **Bromma** 20
Tel: (08) 730 05 50
Telex: 10721
Cable: MEASUREMENTS
Stockholm

Hewlett-Packard Sverige AB
Frötallsgatan 30
S-421 32 **Västra Frölunda**
Tel: (031) 49 09 50
Telex: 10721 via Bromma office

SWITZERLAND
Hewlett-Packard (Schweiz) AG
Zürcherstrasse 20
P.O. Box 307
CH-8952 **Schlieren-Zürich**
Tel: (01) 7305240
Telex: 53933 hpag ch
Cable: HPAG CH

Hewlett-Packard (Schweiz) AG
Château Bloc 19
CH-1219 **Le Lignon-Geneva**
Tel: (022) 96 03 22
Telex: 27333 hpag ch
Cable: HEWPACKAG Geneva

SYRIA
General Electronic Inc.
Nuri Basha-Ahnaf Ebn Kays Street
P.O. Box 5781
Damascus
Tel: 33 24 87
Telex: 11215 ITIKAL
Cable: ELECTROBOR DAMASCUS

Medical only
Sawah & Co.
Place Azmé
B.P. 2308
Damascus
Tel: 16 367-19 697-14 268
Telex: 11304 SATACO SY
Cable: SAWAH, DAMASCUS

Suleiman Hilal El Mlawi
P.O. Box 2528
Mamoun Bitar Street, 56-58
Damascus
Tel: 11 46 63
Telex: 11270
Cable: HILAL DAMASCUS

TAIWAN
Hewlett-Packard Far East Ltd.
Taiwan Branch
Bank Tower, 5th Floor
205 Tun Hau North Road
Taipei
Tel: (02) 751-0404 (15 lines)

Hewlett-Packard Far East Ltd.
Taiwan Branch
68-2, Chung Cheng 3rd. Road
Kaohsiung
Tel: (07) 242318-Kaohsiung

Analytical Only
San Kwang Instruments Co., Ltd.
20 Yung Sui Road
Taipei
Tel: 3615446-9 (4 lines)
Telex: 22894 SANKWANG

TANZANIA
Medical Only
International Aeradio (E.A.), Ltd.
P.O. Box 861
Dar es Salaam
Tel: 21251 Ext. 265
Telex: 41030

THAILAND
UNIMESA Co. Ltd.
Elcom Research Building
2538 Sukumvit Ave.
Bangchak, Bangkok
Tel: 39-32-387, 39-30-338

TRINIDAD & TOBAGO
CARTEL
Caribbean Telecoms Ltd.
P.O. Box 732
69 Frederick Street
Port-of-Spain
Tel: 62-53068

TUNISIA
Tunisie Electronique
31 Avenue de la Liberte
Tunis
Tel: 280 144

Corema
1 ter. Av. de Carthage
Tunis
Tel: 253 821
Telex: 12319 CABAM TN

TURKEY
TEKNIM Company Ltd.
Riza Sah Pehlevi
Caddesi No. 7
Kavaklidere, **Ankara**
Tel: 275800
Telex: 42155

Teknim Com., Ltd.
Barbaros Bulvari 55/12
Besikyas, **Istanbul**
Tel: 613 546
Telex: 23540

E.M.A.
Muhendislik Kollektif Sirketi
Medina Eldem Sokak 41/6
Yüksel Caddesi
Ankara
Tel: 17 56 22

Yilmaz Ozyurek
Milli Mudafaa Cad 16/6
Kizilay
Ankara
Tel: 25 03 09 - 17 80 26
Telex: 42576 OZEK TR

UNITED ARAB EMIRATES
Emitac Ltd. (Head Office)
P.O. Box 1641
Sharjah
Tel: 354121/3
Telex: 8136

Emitac Ltd. (Branch Office)
P.O. Box 2711
Abu Dhabi
Tel: 331370/1

UNITED KINGDOM
Hewlett-Packard Ltd.
King Street Lane
Winnersh, Wokingham
Berkshire RG11 5AR
GB-England
Tel: (0734) 784774
Telex: 84 71 78/9

Hewlett-Packard Ltd.
Fourier House,
257-263 High Street
London Colney
St. Albans, Herts
GB-England
Tel: (0727) 24400
Telex: 1-8952716

Hewlett-Packard Ltd.
Trafalgar House
Navigation Road
Altrincham
Cheshire WA14 1NU
GB-England
Tel: (061) 928 6422
Telex: 668068

Hewlett-Packard Ltd.
Lygon Court
Hereward Rise
Dudley Road
Halesowen,
West Midlands, B62 8SD
GB-England
Tel: (021) 501 1221
Telex: 339105

Hewlett-Packard Ltd.
Wedge House
799, London Road
Thornton Heath
Surrey, CR4 6XL
GB-England
Tel: (01) 684-0103/8
Telex: 946825

Hewlett-Packard Ltd.
14 Wesley St
Castleford
Yorks WF10 1AE
Tel: (0977) 550016
TWX: 5557335

Hewlett-Packard Ltd.
Tradax House
St. Mary's Walk
Maidenhead
Berkshire, SL6 1ST
GB-England

Hewlett-Packard Ltd.
Morley Road
Staplehill
Bristol, BS16 4QT
GB-England

Hewlett-Packard Ltd.
South Queensferry
West Lothian, EH30 9TG
GB-Scotland
Tel: (031) 331 1188
Telex: 72682

UNITED STATES

ALABAMA
700 Century Park South, Suite 128
Birmingham 35226
Tel: (205) 822-6802

P.O. Box 4207
8290 Whitesburg Dr.
Huntsville 35802
Tel: (205) 881-4591

ARIZONA
2336 E. Magnolia St.
Phoenix 85034
Tel: (602) 273-8000

2424 East Aragon Rd.
Tucson 85706
Tel: (602) 273-8000

***ARKANSAS**
Medical Service Only
P.O. Box 5646
Brady Station
Little Rock 72215
Tel: (501) 376-1844

CALIFORNIA
1579 W. Shaw Ave.
Fresno 93771
Tel: (209) 224-0582

1430 East Orangethorpe Ave.
Fullerton 92631
Tel: (714) 870-1000

5400 West Rosecrans Blvd.
P.O. Box 92105
World Way Postal Center
Los Angeles 90009
Tel: (213) 970-7500
TWX: 910-325-6608

3939 Lankershim Boulevard
North Hollywood 91604
Tel: (213) 877-1282
TWX: 910-499-2671

3200 Hillview Av
Palo Alto, CA 94304
Tel: (408) 988-7000

646 W. North Market Blvd.
Sacramento 95834
Tel: (916) 929-7222

9606 Aero Drive
P.O. Box 23333
San Diego 92123
Tel: (714) 279-3200

363 Brookhollow Dr.
Santa Ana, CA 92705
Tel: (714) 641-0977

3003 Scott Boulevard
Santa Clara 95050
Tel: (408) 988-7000
TWX: 910-338-0518

454 Carlton Court
So. San Francisco 94080
Tel: (415) 877-0772

***Tarzana**
Tel: (213) 705-3344

COLORADO
5600 DTC Parkway
Englewood 80110
Tel: (303) 771-3455

CONNECTICUT
47 Barnes Industrial Road
Barnes Park South
Wallingford 06492
Tel: (203) 265-7801

FLORIDA
P.O. Box 24210
2727 N.W. 62nd Street
Ft. Lauderdale 33309
Tel: (305) 973-2600

4080 Woodcock Drive #132
Brownett Building
Jacksonville 32207
Tel: (904) 398-0663

P.O. Box 13910
6177 Lake Ellenor Dr.
Orlando 32809
Tel: (305) 859-2900

P.O. Box 12826
Suite 5, Bldg. 1
Office Park North
Pensacola 32575
Tel: (904) 476-8422
110 South Hoover Blvd.
Suite 120
Tampa 33609
Tel: (813) 872-0900

GEORGIA
P.O. Box 105005
450 Interstate North Parkway
Atlanta 30348
Tel: (404) 955-1500
TWX: 810-766-4890

Medical Service Only
***Augusta** 30903
Tel: (404) 736-0592

P.O. Box 2103
1172 N. Davis Drive
Warner Robins 31098
Tel: (912) 922-0449

HAWAII
2875 So. King Street
Honolulu 96826
Tel: (808) 955-4455

ILLINOIS
211 Prospect Rd.
Bloomington 61701
Tel: (309) 663-0383

5201 Tollview Dr.
Rolling Meadows 60008
Tel: (312) 255-9800
TWX: 910-687-2260

INDIANA
7301 North Shadeland Ave.
Indianapolis 46250
Tel: (317) 842-1000
TWX: 810-260-1797

IOWA
2415 Heinz Road
Iowa City 52240
Tel: (319) 351-1020

KENTUCKY
10170 Linn Station Road
Suite 525
Louisville 40223
Tel: (502) 426-0100

LOUISIANA
P.O. Box 1449
3229-39 Williams Boulevard
Kenner 70062
Tel: (504) 443-6201

MARYLAND
7121 Standard Drive
Parkway Industrial Center
Hanover 21076
Tel: (301) 796-7700
TWX: 710-862-1943

2 Choke Cherry Road
Rockville 20850
Tel: (301) 948-6370
TWX: 710-828-9684

MASSACHUSETTS
32 Hartwell Ave.
Lexington 02173
Tel: (617) 861-8960
TWX: 710-326-6904

MICHIGAN
23855 Research Drive
Farmington Hills 48024
Tel: (313) 476-6400

7224 West Centre Ave.
Kalamazoo 49002
Tel: (616) 323-8362

MINNESOTA
2400 N. Prior Ave.
St. Paul 55113
Tel: (612) 636-0700

MISSISSIPPI
322 N. Mart Plaza
Jackson 39206
Tel: (601) 982-9363

MISSOURI
11131 Colorado Ave.
Kansas City 64137
Tel: (816) 763-8000
TWX: 910-771-2087

1024 Executive Parkway
St. Louis 63141
Tel: (314) 878-0200

NEBRASKA
Medical Only
7101 Mercy Road
Suite 101
Omaha 68106
Tel: (402) 392-0948

NEVADA
***Las Vegas**
Tel: (702) 736-6610

NEW JERSEY
Crystal Brook Professional Building
Route 35
Eatontown 07724
Tel: (201) 542-1384

W. 120 Century Rd.
Paramus 07652
Tel: (201) 265-5000
TWX: 710-990-4951

NEW MEXICO
P.O. Box 11634
Station E
11300 Lomas Blvd., N.E.
Albuquerque 87123
Tel: (505) 292-1330
TWX: 910-989-1185

156 Wyatt Drive
Las Cruces 88001
Tel: (505) 526-2484
TWX: 910-9983-0550

NEW YORK
6 Automation Lane
Computer Park
Albany 12205
Tel: (518) 458-1550
TWX: 710-444-4961

650 Perinton Hill Office Park
Fairport 14450
Tel: (716) 223-9950
TWX: 510-253-0092

No. 1 Pennsylvania Plaza
55th Floor
34th Street & 8th Avenue
New York 10001
Tel: (212) 971-0800

5858 East Molloy Road
Syracuse 13211
Tel: (315) 455-2486

1 Crossways Park West
Woodbury 11797
Tel: (516) 921-0300
TWX: 510-221-2183
Tel: (513) 671-7400

NORTH CAROLINA
5605 Roanne Way
Greensboro 27409
Tel: (919) 852-1800

OHIO
Medical/Computer Only
9920 Carver Road
Cincinnati 45242
Tel: (513) 891-9870
16500 Sprague Road
Cleveland 44130
Tel: (216) 243-7300
TWX: 810-423-9430

962 Crupper Ave.
Columbus 43229
Tel: (614) 436-1041

330 Progress Rd.
Dayton 45449
Tel: (513) 859-8202

OKLAHOMA
P.O. Box 32008
6301 N. Meridan Avenue
Oklahoma City 73112
Tel: (405) 721-0200

9920 E. 42nd Street
Suite 121
Tulsa 74145
Tel: (918) 665-3300

OREGON
17890 S.W. Lower Boones Ferry Road
Tualatin 97062
Tel: (503) 620-3350

PENNSYLVANIA
1021 8th Avenue
King of Prussia Industrial Park
King of Prussia 19406
Tel: (215) 265-7000
TWX: 510-660-2670

111 Zeta Drive
Pittsburgh 15238
Tel: (412) 782-0400

SOUTH CAROLINA
P.O. Box 6442
6941-0 N. Trenholm Road
Columbia 29206
Tel: (803) 782-6493

TENNESSEE
8906 Kingston Pike
Knoxville 37919
Tel: (615) 691-2371

3070 Directors Row
Directors Square
Memphis 38131
Tel: (901) 346-8370

***Nashville**
Medical Service Only
Tel: (615) 244-5448

TEXAS
4171 North Mesa
Suite C110
El Paso 79902
Tel: (915) 533-3555

P.O. Box 42816
10535 Harwin St.
Houston 77036
Tel: (713) 776-6400

***Lubbock**
Medical Service Only
Tel: (806) 799-4472

P.O. Box 1270
201 E. Arapaho Rd.
Richardson 75081
Tel: (214) 231-6101

205 Billy Mitchell Road
San Antonio 78226
Tel: (512) 434-8241

UTAH
2160 South 3270 West Street
Salt Lake City 84119
Tel: (801) 972-4711

VIRGINIA
P.O. Box 9669
2914 Hungary Spring Road
Richmond 23228
Tel: (804) 285-3431

Computer Systems/Medical Only
Airport Executive Center
Suite 302
5700 Thurston Avenue
Virginia Beach 23455
Tel: (804) 460-2471

WASHINGTON
Bellefield Office Pk.
1203 - 114th Ave. S.E.
Bellevue 98004
Tel: (206) 454-3971
TWX: 910-443-2446

P.O. Box 4010
Spokane 99202
Tel: (509) 535-0864

***WEST VIRGINIA**
Medical/Analytical Only
4604 Mac Corkle Ave., S.E.
Charleston 25304
Tel: (304) 925-0492

WISCONSIN
150 South Sunny Slope Road
Brookfield 53005
Tel: (414) 784-8800

FOR U.S. AREAS NOT LISTED:
Contact the regional office nearest you:
Atlanta, Georgia . . . **North Holly-wood,** California . . . **Rockville,** Maryland . . . **Rolling Meadows,** Illinois.
Their complete addresses are listed above.

USSR
Hewlett-Packard Representative Office
USSR
Pokrovsky Boulevard
4/17-kw 12
Moscow 101000
Tel: 294.20.24
Telex: 7825 hewpak su

YUGOSLAVIA
Iskra Commerce, n.sol.o.
Zastopstvo Hewlett-Packard
Obilicev Venac 26
YU 11000 **Beograd**
Tel: 636-955
Telex: 11530

Iskra Commerce, n.sol.o.
Zastopstvo Hewlett-Packard
Miklosiceva 38/VII
YU-61000 **Ljubljana**
Tel: 321-674, 315-879
Telex: 31583

URUGUAY
Pablo Ferrando S.A.C.el.
Avenida Italia 2877
Casilla de Correo 370
Montevideo
Tel: 40-3102
Telex: 702 Public Booth
Para Pablo Ferrando

VENEZUELA
Hewlett-Packard de Venezuela C.A.
P.O. Box 50933
Caracas 105
Los Ruices Norte
3a Transversal
Edificio Segre
Caracas 107
Tel: 239-4133 (20 lines)
Telex: 25146 HEWPACK

ZAMBIA
R.J. Tilbury (Zambia) Ltd.
P.O. Box 2792
Lusaka
Tel: 73793

MEDITERRANEAN AND MIDDLE EAST COUNTRIES NOT SHOWN, PLEASE CONTACT:
Hewlett-Packard S.A.
Mediterranean and Middle East Operations
35, Kolokotroni Street
Platia Kefallariou
GR-Kifissia-**Athens,** Greece
Tel: 8080359/429
Telex: 21-6588
Cable: HEWPACKSA Athens

SOCIALIST COUNTRIES NOT SHOWN, PLEASE CONTACT:
Hewlett-Packard Ges.m.b.H.
Handelskai 52
P.O. Box 7
A-1205 **Vienna,** Austria
Tel: (0222) 35 16 21 to 27
Cable: HEWPAK Vienna
Telex: 75923 hewpak a

OTHER AREAS NOT LISTED, CONTACT:
Hewlett-Packard Intercontinental
3495 Deer Creek Road
Palo Alto, California 94304
Tel: (415) 856-1501
TWX: 910-373-1267
Cable: HEWPACK Palo Alto
Telex: 034-8300, 034-8493

Hewlett-Packard S.A.
7, rue du Bois-du-Lan
P.O. Box
CH-1217 Meyrin 2 - **Geneva**
Switzerland
Tel: (022) 82 70 00
Cable: HEWPACKSA Geneva
Telex: 2 24 86

***Service Only**
2-15-80

Part No. 30000-90042
Printed in U.S.A. 6/78
Update No. 2 Incorporated 4/79

HEWLETT PACKARD